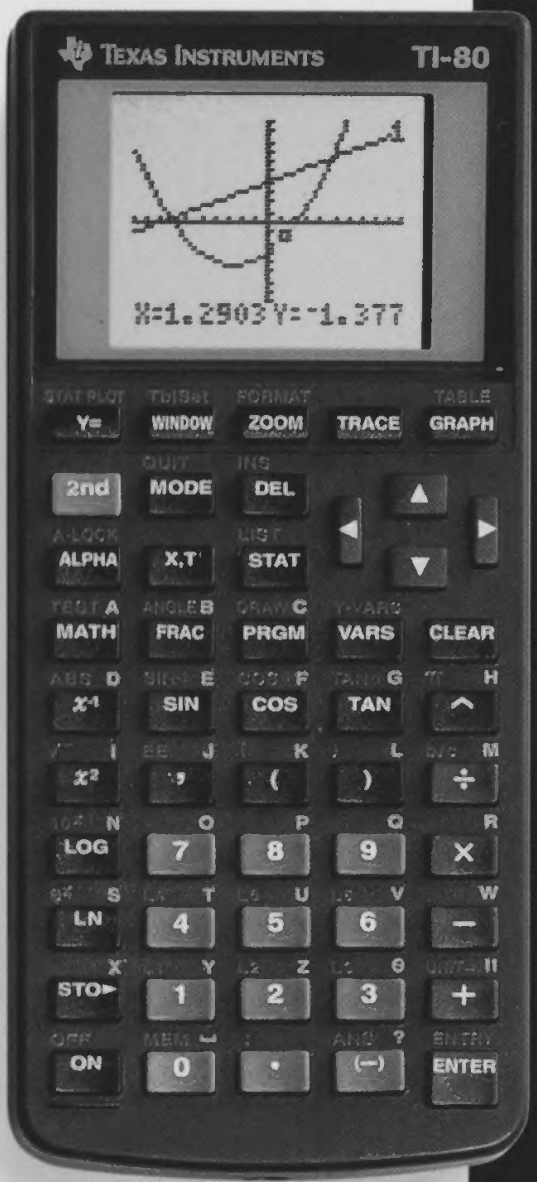
# TI-80 Graphic Calculator

# USER MANUAL




TEXAS INSTRUMENTS

TI-80 User Manual

GB

TEXAS INSTRUMENTS


# TI-80

## OWNER'S MANUAL

**Important**

Texas Instruments makes no warranty, either expressed or implied, including but not limited to any implied warranties of merchantability and fitness for a particular purpose, regarding any programs or book materials and makes such materials available solely on an "as-is" basis.

In no event shall Texas Instruments be liable to anyone for special, collateral, incidental, or consequential damages in connection with or arising from the purchase or use of these materials, and the sole and exclusive liability of Texas Instruments, regardless of the form of action, shall not exceed the purchase price of this equipment. Moreover, Texas Instruments shall not be liable for any claim of any kind whatsoever against the use of these materials by any other party.

**Caution**: Any changes or modifications to this equipment not expressly approved by Texas Instruments may void your authority to operate the equipment.

# TEXAS INSTRUMENTS

## WARRANTY CARD

## TWO (2) YEAR WARRANTY

**Name :**

**Profession :**

**Address :**

**Telephone :**

- Warranty is accepted if this card is correctly filled in by the retailer: stamp & date of purchase are required. If these are missing, you must send in the invoice or your receipt proving the purchase.).

- To simplify your return and expedite its processing, please complete the other side of this card.

Thank you

**Model :**

**TI-80**

**Date of purchase :**

**RETAILER'S STAMP**

ANY DAMAGE TO THE DISPLAY IS NOT COVERED BY WARRANTY.

## IN CASE OF DIFFICULTY

In case of difficulty, please carefully read the information in your manual, calculator and duplicate examples. Please also refer to the section "In Case of Difficulty":

1. In case of difficulty with the display (blank display or digits and graphs do not appear), check the display contrast: Press [ON] to switch the calculator on. Press then release the 2nd. Press and hold the [▲] or [▼] cursor keys to adjust the contrast.

2. In case of erratic functioning or erratic display, reset your calculator. Press [2nd] [OFF] then [ON] to switch the calculator on. Press then release [2nd], then [MEM]. Press [3] (RESET), then press [2] (RESET). Then adjust the contrast (see 1).

3. Checking the batteries: try again with new batteries.

   **Warning:** battery replacement is not covered by warranty.

### REASON FOR RETURN

**DISPLAY** ❑ Dark ❑ No display ❑ Other

Describe the problem : .....

.....

**KEYBOARD**

Indicate which keys are not working : .....

.....

**RANDOM FUNCTIONING :** Wrong calculations, wrong display...

Please give some examples : .....

.....

**OTHERS :**

Please specify : .....

.....

# Table of Contents

**This manual describes how to use the TI-80 Graphics Calculator. Getting Started gives a quick overview of its features. The first chapter gives general instructions on operating the TI-80. Other chapters describe its interactive features. The applications in Chapter 11 show how to use these features in combination.**

# Table of Contents (Continued)

# Table of Contents (Continued)

# Using this Guidebook Effectively

**The structure of the TI-80 guidebook and the design of its pages can help you find the information you need, quickly. Consistent presentation techniques are used throughout to make the guidebook easy to use.**

## Structure of the Guidebook

The guidebook is designed to teach you how to use the calculator.

- Getting Started is a fast-paced, keystroke-by-keystroke introduction.
- Chapter 1 describes general operation and lays the foundations for Chapters 2 to 10, which describe specific functional areas of the TI-80. Most chapters begin with a brief Getting Started introduction.
- Chapter 11 contains application examples that incorporate features from different functional areas of the calculator. These examples can help you see how different functional areas work together to accomplish meaningful tasks.
- Chapter 12 describes memory management.

## Page-Design Conventions

When possible, units of information are presented on a single page or on two facing pages. Several page-design elements help you find information quickly.

- **Page headings**—The descriptive heading at the top of the page or two-page unit identifies the subject of the unit.
- **General text**—Just below the page heading, a short section of bold text provides general information about the subject covered in the unit.
- **Left-column subheadings**—Each subheading identifies a specific topic or task related to the page or unit subject.
- **Specific text**—The text to the right of a subheading presents detailed information about that specific topic or task. The information may be presented as paragraphs, numbered procedures, bulleted lists, or illustrations.
- **Page "footers"**—The bottom of each page shows the chapter name, chapter number, and page number.

### Information-Mapping Conventions

Several conventions are used to present information concisely and in an easily referenced format.

- **Numbered procedures**—A procedure is a sequence of steps that performs a task. In this guidebook, each step is numbered in the order in which it is performed. No other text in the guidebook is numbered; therefore, when you see numbered text, you know you should perform the steps sequentially.
- **Lists with bullets**—If several items have equal importance, or if you may choose from one of several alternative actions, this guidebook precedes each item with a "bullet" (•) to highlight it—like this list.
- **Tables and charts**—Sets of related information are presented in tables or charts for quick reference.
- **Keystroke examples**—The Getting Started examples provide keystroke-by-keystroke instructions, as do the numerous short examples and several detailed examples, identified with a  .

### Reference Aids

Several techniques have been used to help you look up specific information when you need it. These include:

- A chapter table of contents on the first page of each chapter, as well as the full table of contents at the front of the guidebook.
- A glossary at the end of this section, defining important terms used throughout the guidebook.
- An alphabetical table of functions and instructions in Appendix A, showing their correct formats, how to access them, and page references for more information.
- Information about system variables in Appendix A.
- A table of error messages in Appendix B, showing the messages and their meanings and giving problem-handling information.
- An alphabetical index at the back of the guidebook, listing tasks and topics you may need to look up.

# Glossary

**This glossary provides definitions for important terms that are used throughout this guidebook.**

**Argument** — An argument is an input item upon which the value of a function depends.

**Command** — A command is any entry submitted to the calculator using [ENTER]. There are two types of TI-80 command: instructions and expressions.

**Expression** — An expression is a complete sequence of numbers, variables, functions, and their arguments that can be evaluated to a single answer. An expression returns the evaluated result to **ANS**.

**Function** — A function, which may have arguments, returns a value and can be used in an expression.

A function is also the expression entered in the **Y=** editor used in graphing.

**Home Screen** — The Home screen is the primary screen of the TI-80, where expressions can be entered and evaluated and instructions can be entered and executed.

**Instruction** — An instruction, which may have arguments, initiates an action. Instructions are not valid in expressions. An instruction does not return a value to **ANS**.

**List** — A list is a set of values that the TI-80 can use for activities such as evaluating a function at multiple values and entering statistical data.

**Menu Items** — Menu items are shown on full-screen menus.

**Pixel** — A pixel (picture element) is a square dot on the TI-80 display. The TI-80 display is 64 pixels wide and 48 pixels high.

**Real Number** — On the TI-80, real numbers are individual decimal or fraction values.

**Value** — A value is a single decimal or fraction number or a list of decimals or fractions.

**Variable** — A variable is the name given to a location in memory in which a value, an expression, a list, or another named item is stored.

# Getting Started: Do This First!

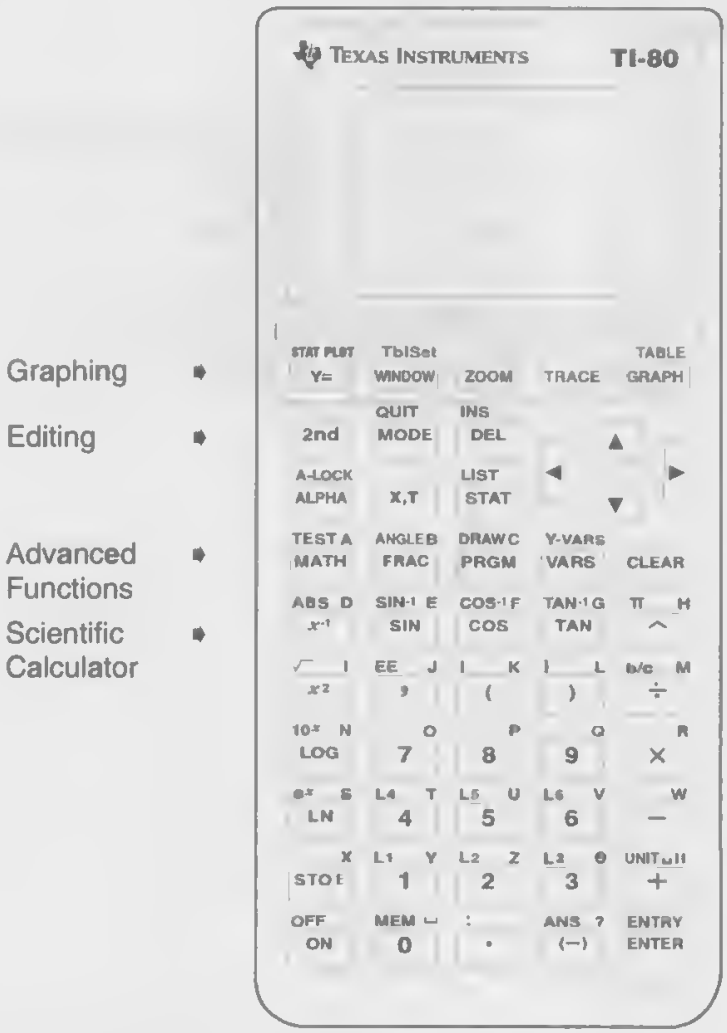
**Getting Started contains two keystroke-by-keystroke examples—an interest rate problem and a volume problem—which introduce you to some of the principal operating and graphing features on the TI-80. You will learn to use the TI-80 more quickly by completing both of these examples first.**

**Contents**

# TI-80 Keyboard

**The keys on the TI-80 are grouped by colour and physical layout to allow easy location of the key you need. The keys are divided into zones: graphing keys, editing keys, advanced function keys, and scientific calculator keys.**

## The Zones of the Keyboard




**Graphing Keys** These keys are most frequently used to access the interactive graphing features of the TI-80.

**Editing Keys** These keys are most frequently used for editing expressions and values.

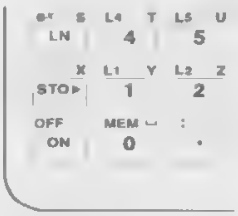
**Advanced Function Keys** These keys are most frequently used to access the advanced functions of the TI-80.

**Scientific Calculator Keys** These keys are most frequently used to access the functions of a standard scientific calculator.

**Before beginning the two sample problems, follow the steps on this page to reset the TI-80 to its factory settings. (Resetting the TI-80 erases all previously entered data.) This ensures that following the keystrokes in this section will produce the same actions.**

1. Press [ON] to turn the calculator on.

   If the screen is very dark or blank, adjust the display contrast. Press and release [2nd], and then press and hold [▼] (to make the display lighter) or press and hold [▲] (to make the display darker). You can press [CLEAR] to clear the display.




2. Press and release [2nd], and then press [0]. (Pressing [2nd] gives you access to the **2nd** operations, which are printed above the keys on the left. **MEM** is the **2nd** operation of the [0] key.)

   The **MEMORY** menu is displayed.


3. Press **3** to select **RESET...**.

   The **MEMORY RESET** menu is displayed.


4. Press **2** to select **RESET**. The calculator is reset.

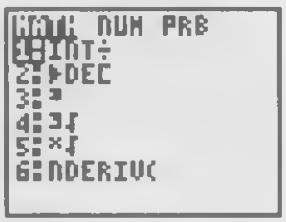
# TI-80 Menus

**To leave the keyboard uncluttered, the TI-80 uses full-screen menus to display many additional operations. The use of specific menus is described in the appropriate chapters.**

## Displaying a Menu

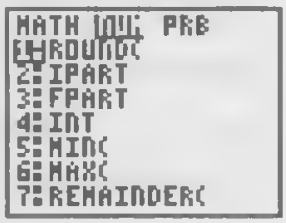
When you press a key that displays a menu, such as MATH, that menu screen temporarily replaces the screen you are working on.

After you make a selection from a menu, you are usually returned to the screen where you were, previously.




## Moving from One Menu to Another

A menu key may display more than one menu name. The names appear on the top line. The name of the current menu is highlighted, and the items in that menu are displayed. Use ▶ or ◀ to display a different menu.



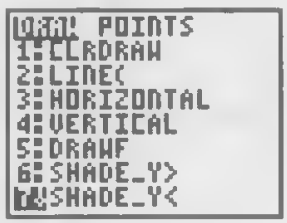

## Selecting an Item from a Menu

The number of the current item is highlighted. If there are more than seven items on the menu, a ↓ appears on the last line in place of the : (colon).

To select from a menu, you can either:

- Use ▼ and ▲ to move the cursor to the item, and then press ENTER, or
- Press the number of the item.

**Note:** The tenth item in a menu is number **0**. If there are more than 10 items, they are numbered A, B, C, etc. To select one of these items, press ALPHA and then the letter.




## Leaving without Making a Selection

To leave a menu without making a selection:

- Press 2nd [QUIT] to return to the Home screen.
- Press CLEAR to return to the screen where you were.
- Press the key for another screen or menu.

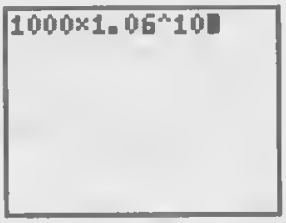
# Entering a Calculation: Compound Interest

**The TI-80 displays up to 8 16-character lines so that you see an expression and its solution together. You can store values to variables, enter multiple instructions on one line, and recall previous entries.**

**By trial and error, determine when £1000 invested at 6% annual compound interest will double in value.**

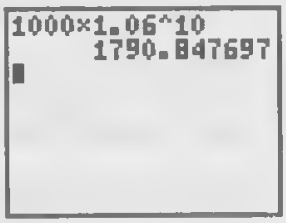
1. For the first guess, calculate the amount available at the end of 10 years. Enter the expression just as you would write it.

   Press **1000** [×] **1.06** [^] **10**.

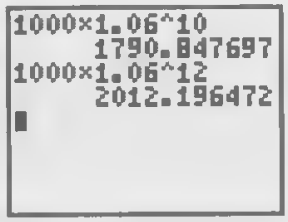



2. Press [ENTER] to evaluate the expression.

   The answer is shown on the right-hand side of the display. The cursor is positioned on the next line, ready for you to enter the next expression.




3. The next guess should be greater than 10 years. Make the next guess 12 years. To calculate the amount after 12 years, press **1000** [×] **1.06** [^] **12**, followed by [ENTER].

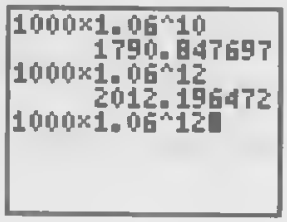


## Continuing a Calculation

**To save keystrokes, you can use the Last Entry feature to recall the last expression entered and then edit it for a new calculation. In addition, the next expression can be continued from the previous result.**

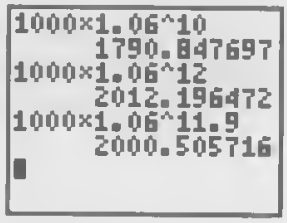
1. The next guess should be less than, but close to, 12 years. Calculate the amount available at the end of 11.9 years, using the Last Entry feature. Press [2nd], followed by [ENTRY] (the second function of [ENTER]).

   The last calculated expression is shown on the next line of the display. The cursor is positioned at the end of the expression.




2. You can edit the expression. Press [◀] to move the cursor over the **2**. Then type **1.9** to change **12** to **11.9**. Press [ENTER] to evaluate the expression.

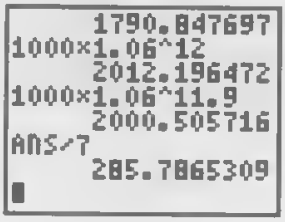
   **Note:** This process can be continued to obtain a solution with the desired accuracy.




3. You can continue a calculation using the result of the last calculation. For example, if the final amount determined above is to be divided among seven people, how much would each person get?

   To divide the last calculation by seven, press [÷] **7**, followed by [ENTER].

   As soon as you press [÷], **ANS/** is displayed at the beginning of the new expression. **ANS** is a variable that contains the last calculated result. In this case, **ANS** contains the variable 2000.505716.



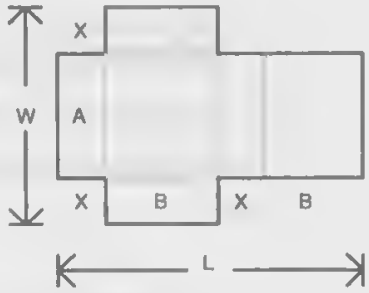
# Defining a Function: Box with Lid

**Take an 21.0 cm × 29.7 cm sheet of paper and cut X by X squares from two corners and X by (X+B) rectangles from the other two corners. Now fold the paper into a box with lid. What X would give the maximum volume V of a box made in this way? Use tables and graphs to determine the solution.**

Begin by defining a function that describes the volume of the box.

From the diagram: $2X + A = W$
$2X + 2B = L$
$V = A\ B\ X$

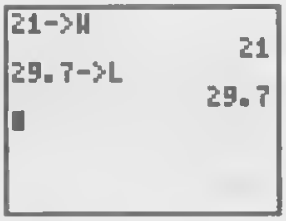
Substituting: $V = (W - 2X)(L/2 - X)X$




If necessary, press [MODE] [▼] [ENTER] to change the **MODE** to **FLOAT**. Then press [2nd] [Quit] [CLEAR] to return to the Home screen and clear it.

1. Press **21** [STO▸] [ALPHA] **W** [ENTER] to store the width of the paper.

   Press **29.7** [STO▸] [ALPHA] **L** [ENTER] to store the length of the paper.

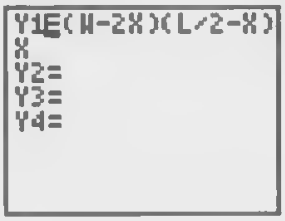



2. You can define functions for tables and graphs on the **Y=** edit screen.

   Press [Y=] to access this screen.


3. Enter the function for volume as **Y1**. Press [(] [ALPHA] **W** [−] **2** [X,T] [)] [(] [ALPHA] **L** [÷] **2** [−] [X,T] [)] [X,T] [ENTER] to define function **Y1** in terms of **X**. ([X,T] lets you quickly enter **X** without pressing [ALPHA].)

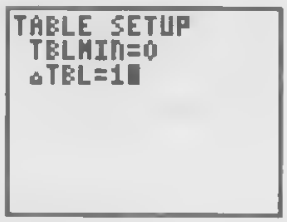
   The = sign is highlighted to show that **Y1** is selected.



# Defining a Table of Values

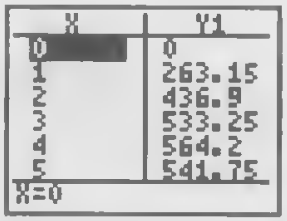
**The table feature of the TI-80 provides numeric information about a function. Use a table of values from the previously defined function to estimate an answer to the problem.**

1. Press [2nd] [TblSet] (above [WINDOW]) to display the **TABLE SETUP** screen.
2. Press [ENTER] to accept **TBLMIN=0**.
3. Press **1** [ENTER] to define the table increment Δ **TBL=1**.

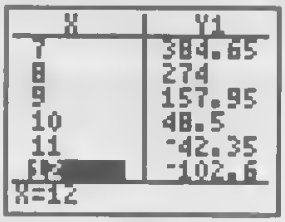



4. Press [2nd] [TABLE] (above [GRAPH]) to display the table.

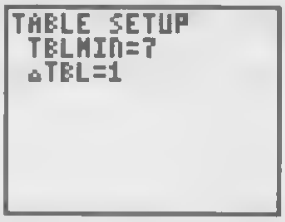
   Note that the maximum value is around **4**, between **3** and **5**.




5. Press and hold [▼] to scroll through the table until the sign change appears. Note that the maximum length of **X** for this problem occurs where the sign of **Y1** (volume) becomes negative.




6. Press [2nd] [TblSet]. Note that **TBLMIN** has changed to reflect the first line of the table you last displayed.

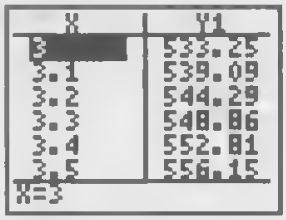


## Zooming In on the Table

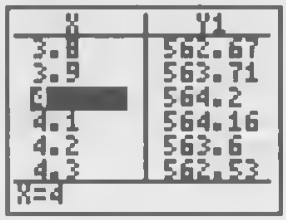
**You can adjust the way a table is displayed to get more detailed information about any defined function. By varying the value of ΔTBL, you can "zoom in" on the table.**

1. Adjust the table setup to get a more accurate estimate of the maximum size of the cutout. Press **3** [ENTER] to set **TBLMIN**. Press **.1** to set **ΔTBL**.

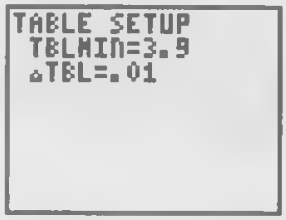

2. Press [2nd] [TABLE].



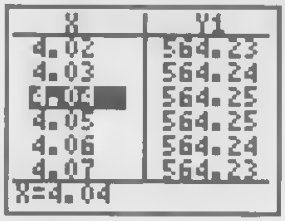

3. Use [▼] and [▲] to scroll through the table. Note that the maximum value displayed is **564.2**, which occurs at **X=4**. The maximum occurs at **3.9<X<4.1**.



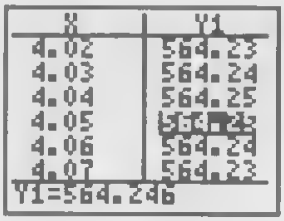
4. Press [2nd] [TblSet]. Press **3.9** [ENTER] to set **TBLMIN**. Press **.01** [ENTER] to set **ΔTBL**.



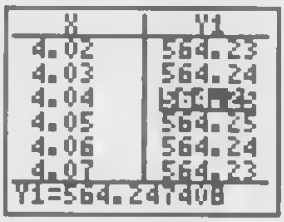

5. Press [2nd] [TABLE] and use [▼] and [▲] to scroll through the table. Two "equal" maximum values are shown, **564.25** at **X=4.04** and **X=4.05**.




6. Press [▼] or [▲] to move the cursor to **4.05**. Press [▶] to move the cursor into the **Y1** column. The bottom line of the display shows the value of **Y1** at **4.05** in full precision, **564.246**.



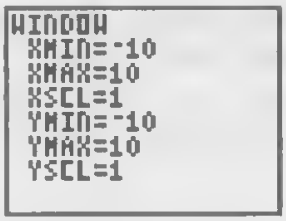

7. Press [▼] to display the "other" maximum. The value of **Y1** at **4.04** in full precision is **564.247408**. This would be the maximum volume of the box if you could cut your piece of paper at .01-cm increments.



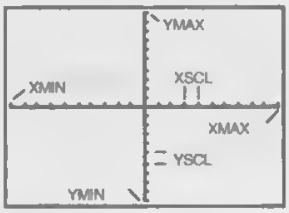
## Changing the Viewing Window

**The viewing window defines the portion of the coordinate plane that appears in the display. The values of the Window variables determine the size of the viewing window. You can view and change these values.**

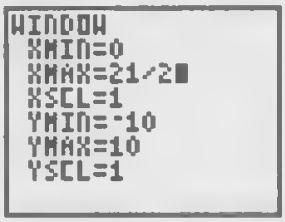
1. Press [WINDOW] to display the Window variables edit screen. You can view and edit the values of the Window variables here.



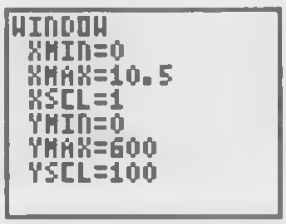

The standard Window variables define the viewing window as shown. **XMIN**, **XMAX**, **YMIN**, and **YMAX** define the boundaries of the display. **XSCL** and **YSCL** define the distance between tick marks on the **X** and **Y** axes.




2. Press **0** [ENTER] to define **XMIN**.
3. You can enter expressions to define values in the window editor. Press **21** [÷] **2**.




4. Press [ENTER]. The expression is evaluated, and **10.5** is stored in **XMAX**. Press [ENTER] to accept **XSCL** as **1**.
5. Press **0** [ENTER] **600** [ENTER] **100** [ENTER] to define the **Y** Window variables.

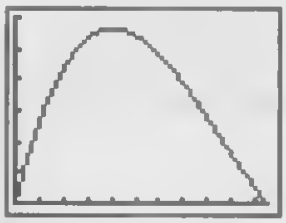


## Displaying and Tracing the Graph

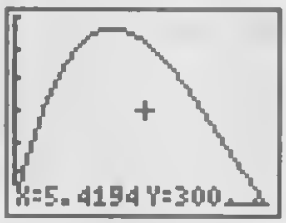
**Now that you have defined the function to be graphed and the window in which to graph it, you can display and explore the graph. You can trace along a function with TRACE.**

1. Press [GRAPH] to graph the selected function in the viewing window.

   The graph of **Y1=(W–2X)(L/2–X)X** is shown in the display.

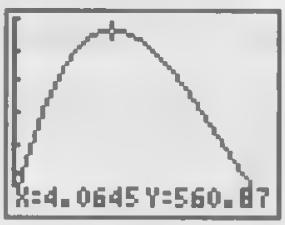


2. Press [▶] once to display the free-moving cursor just to the right of the centre of the screen. The bottom line of the display shows the X- and Y-coordinate values for the position of the graph cursor.



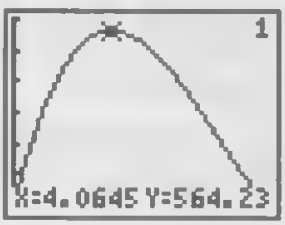

3. Use [◀], [▶], [▲], and [▼] to position the free-moving cursor at the apparent maximum of the function.

   As you move the cursor, the X- and Y-coordinate values are updated continually to reflect the cursor position.




4. Press [TRACE]. The Trace cursor appears on the **Y1** function. **1** in the upper right corner of the display shows that the cursor is on **Y1**. As you press [◀] and [▶], you trace along **Y1**, one **X** dot at a time, evaluating **Y1** at each **X**.

5. Press [◀] and [▶] until you are at the maximum Y value. This is the maximum of **Y1(X)** for the X pixels. (There may be a maximum value "in between" pixels.)

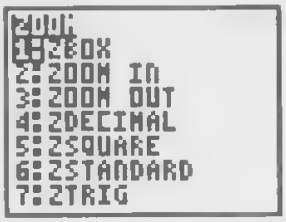


# Zooming In on the Graph

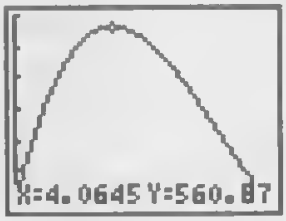
**You can magnify the viewing window around a specific location using the Zoom instructions to help identify maximums, minimums, roots, and intersections of functions.**

1. Press [ZOOM] to display the **ZOOM** menu.

   This menu is typical of TI-80 menus. To select an item, you may either press the number to the left of the item, or you may press [▼] until the item number is highlighted, and then press [ENTER].

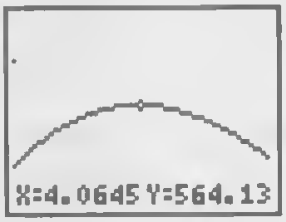



2. To **ZOOM IN**, press **2**. The graph is displayed again. The cursor changes to indicate that you are using a Zoom instruction.

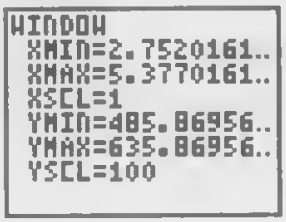



3. Use [◄], [▲], [►], and [▼] to position the cursor near the maximum value on the function, and press [ENTER].

   The new viewing window is displayed. It has been adjusted in both the **X** and **Y** directions by factors of 4 which are the values for Zoom factors.




4. Press [WINDOW] to display the new window variable values.



# Other TI-80 Features

**Getting Started has introduced you to the basic calculator operations and the table and function graphing features of the TI-80. The remainder of this Guidebook describes these features in more detail and also covers other functions of the TI-80.**

| | |
|---|---|
| **Fractions** | You can enter fractions directly from the keyboard and perform calculations with fractions. You can convert between fractions and their decimal equivalents. In **MANSIMP** mode, you can simplify fractions, step-by-step. The TI-80 tells you when a fraction can be simplified and shows the common factor after simplification (Chapter 3). |
| **Graphing** | You can store, graph, and analyse up to four functions (Chapter 4) and up to three parametric functions (Chapter 5). You can use Draw operations to annotate graphs (Chapter 7). |
| **Tables** | You can create function evaluation tables to simultaneously analyse multiple functions (Chapter 6). |
| **Lists** | You can enter and save up to six lists for use in statistical analysis. You also can use lists to simultaneously evaluate expressions at multiple values (Chapter 8). |
| **Statistics** | You can perform one-variable and two-variable list-based statistical analysis, including regression analysis, and plot the data as histograms, points, x-y lines, or box-and-whisker plots. You can define and save three statistical plot definitions (Chapter 9). |
| **Programming** | You can enter and save programs that include extensive control and input/output instructions (Chapter 10) |

# Chapter 1: Operating the TI-80

**This chapter describes the TI-80 and provides general information about its operation.**

**Chapter Contents**

# Turning the TI-80 On and Off

**To turn the TI-80 on, press the [ON] key. To turn it off, press and release [2nd], and then press [OFF]. After about five minutes without any activity, the APD™ (Automatic Power Down) feature turns the TI-80 off automatically.**

### Turning the Calculator On

Press [ON] to turn the TI-80 on.

- If you pressed [2nd] [OFF] to turn the calculator off, the display shows the Home screen as it was when you last used it, and errors are cleared.
- If APD turned the calculator off, the display returns to the same screen or editor in which you left it. See "APD (Automatic Power Down)" below.

### Turning the Calculator Off

Press and release [2nd], and then press [OFF] to turn the TI-80 off.

- Any error condition is cleared.
- All settings and memory contents are retained by the Constant Memory™ feature.

### APD (Automatic Power Down)

To prolong the life of the batteries, APD turns the TI-80 off automatically after several minutes without any activity. When you press [ON], the calculator shows the same screen or editor in which you left it.

- If an error message was displayed when APD turned the TI-80 off, the error is cleared and the display returns to a blank line on the Home screen.
- If a menu was displayed, the display returns to the screen or editor from which you called the menu.

All settings and memory contents are retained by the Constant Memory feature.

**Note:** APD does not occur if a calculation or program is in progress, unless the program is paused.

### Batteries

The TI-80 uses two lithium batteries. To replace the batteries without losing any information stored in memory, follow the directions in Appendix B.

# Setting the Display Contrast

**The brightness and contrast of the display depend on room lighting, battery freshness, viewing angle, and the adjustment of the display contrast. The contrast setting is retained in memory when the TI-80 is turned off.**

## Adjusting the Display Contrast

You can adjust the display contrast to suit your viewing angle and lighting conditions. As you adjust the contrast setting, the display becomes lighter or darker. A highlighted number in the top right corner changes to indicate the current contrast setting; 0 is the lightest, and 9 is the darkest.

To adjust the display contrast:

1. Press and release the [2nd] key.
2. Use one of two keys:
   - To increase the contrast (darken the screen), press and hold [▲].
   - To decrease the contrast (lighten the screen), press and hold [▼].

Note: If you adjust the contrast setting too low, the display may become completely blank. If this happens, press and release [2nd], and then press and hold [▲] until the display reappears.

## When to Replace Batteries

As you use the TI-80, the battery voltage will gradually drop, and the display will dim. You can adjust the contrast to darken the display when this happens. If the display is dim and adjusting the contrast does not make it dark enough, you should replace the batteries soon. Refer to Appendix B for instructions on how to change the batteries.

Note: The display contrast may appear very dark after you change batteries. Press and release [2nd], and then press and hold [▼] to lighten the display.

**The TI-80 displays both text and graphs. Graphs are described in Chapters 4 and 5.**

## Home Screen

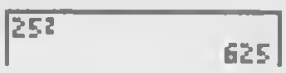
The primary screen of the TI-80 is the Home screen. You enter instructions to be executed, expressions to be evaluated, and see the results on the Home screen.

## Displaying Entries and Answers

When text is displayed, the TI-80 screen can show a maximum of eight lines with 16 characters each.

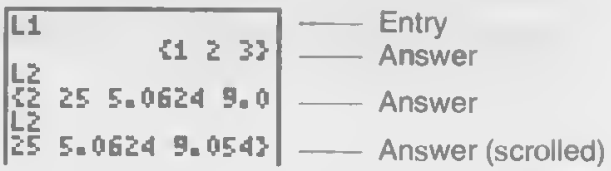
- If all lines of the display are filled, text "scrolls" off the top of the display.
- If an expression on the Home screen, the **Y=** editor (Chapter 4), or the program editor (Chapter 10) is longer than one line, it wraps to the beginning of the next line.
- On numeric editors such as the **WINDOW** screen (Chapter 4), expressions scroll to the left and right.

When an entry is executed on the Home screen, the answer is displayed on the right side of the next line.




The mode settings (pages 1-9 to 1-11) control the way the calculator interprets expressions and displays answers.

If an answer is too long to display in its entirety, you can press ▶ and ◀ to scroll the answer so that you can view all of it. In the second example below, the open brace without a corresponding close brace indicates that the list is too long to be displayed in its entirety.




## Returning to the Home Screen

To return to the Home screen from any other screen, press [2nd] [QUIT].

## Display Cursors

In most cases, the appearance of the cursor indicates what will happen when you press the next key.

| Cursor | Appearance | Meaning |
|---|---|---|
| Entry | Flashing ■ | The next keystroke is entered at the cursor; it types over any character. |
| **INS** (insert) | Flashing _ | The next keystroke is inserted at the cursor. |
| **2nd** | Flashing ↑ | The next keystroke is a **2nd** operation. |
| **ALPHA** | Flashing A | The next keystroke is an alphabetic character. |
| memory "full" | Checked rectangle | You have entered the maximum number of characters in a name, or memory is full. |

Graphs and the screens for viewing and editing tables and lists have different cursors, which are described in the appropriate chapters.

## Busy Indicator

When the TI-80 is calculating or graphing, a vertical line shows in the top right of the display as a busy indicator. During a pause in a program, the busy indicator is a dotted line.

**In most places where a value is required, you can use an expression to enter the value. You can enter instructions, which initiate an action, on the Home screen or in the program editor (Chapter 10).**

## Expressions

An expression is a complete sequence of numbers, variables, functions, and their arguments that evaluate to a single answer. For example, $\pi r^2$ is an expression. On the TI-80, you enter an expression in the same order as you would write it.

You can create expressions on the Home screen to calculate an answer. In most places where a value is required, you can use an expression to enter the value.

## Entering an Expression

To create an expression, you enter numbers, variables, and functions from the keyboard and menus. An expression is completed when you press [ENTER], regardless of the cursor location. The entire expression is evaluated according to Equation Operating System (EOS™) rules, and then the answer is displayed.

**Note:** EOS rules determine the order in which operations are completed (page 1-20).

Most TI-80 functions and operations are symbols with several characters in them. You must enter the symbol from the keyboard or menu. You cannot spell it out. For example, to calculate the log of 45, you must press [LOG] **4 5**. You cannot type in the letters **L O G**. (If you type **LOG**, the TI-80 interprets the entry as implied multiplication of the variables **L**, **O**, and **G**.)

Calculate $3.76 \div (-7.9 + \sqrt{5}) + 2 \log 45$.

**3.76** [÷] [(] [(-)] **7.9** [+] [2nd] [√] **5** [)] [+] **2** [LOG] **45** [ENTER]

```
3.76/(-7.9+√(5))+2
LOG 45
      2.642575252
```

## Multiple Entries on a Line

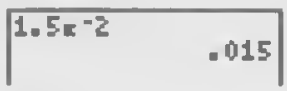
To enter more than one expression or instruction on a line, separate them with a colon (:). They are all stored together in Last Entry (page 1-15).

```
5->A:2->B:A/B
             2.5
```

### Entering a Number in Scientific Notation

To enter a number in scientific notation:

1. Type the part of the number that precedes the exponent. This value can be an expression.
2. Press [2nd] [EE]. E is displayed.
3. If the exponent is negative, press [(-)], and then type the exponent, which can be one or two digits.




Entering a number in scientific notation does not cause the answers to be displayed in scientific notation. The display format is determined by the mode settings (pages 1-9 to 1-11) and the size of the number.

### Functions

A function returns a value. For example, ÷, −, +, √, and **LOG** are functions. Some functions take more than one argument, which is indicated by a **(** at the end of the name. **MIN(** requires two arguments in this example : **MIN(5,8)**.

### Instructions

An instruction initiates an action. For example, **CLRDRAW** is an instruction that clears any drawn elements from a graph. Instructions cannot be used in expressions. Some instructions require more than one argument, as indicated by a **(** at the end of the name. **LINE(** requires four arguments in this example : **LINE(1,1,3,3)**.

### Interrupting a Calculation

While the busy indicator is displayed, indicating that a calculation or a graph is in progress, you can press [ON] to stop the calculation. (There may be a delay.)

# The Edit Keys

**The arrow keys near the top right of the keyboard control the movement of the cursor. In normal entry, a keystroke types over the character or characters at the position of the cursor. The [DEL] and [2nd] [INS] keys delete or insert characters.**

| Key(s) | Action(s) |
|---|---|
| [▶] or [◀] | Moves the cursor within an expression. These keys repeat when you hold them down. |
| [▲] or [▼] | Moves the cursor between lines within an expression. These keys repeat when you hold them down.<br>• On the top line of an expression on the Home screen, [▲] moves the cursor to the beginning of the expression.<br>• On the bottom line of an expression on the Home screen, [▼] moves the cursor to the end of the expression. |
| [2nd] [◀] | Moves the cursor to the beginning of an expression. |
| [2nd] [▶] | Moves the cursor to the end of an expression. |
| [ENTER] | Evaluates an expression or executes an instruction. |
| [CLEAR] | • On a line with text on the Home screen, clears (blanks) the current line.<br>• On a blank line on the Home screen, clears everything on the Home screen.<br>• In an editor, clears (blanks) the expression or value where the cursor is located; it does not store a zero. |
| [DEL] | Deletes the character at the cursor. This key repeats. |
| [2nd] [INS] | Lets you insert characters at the underline cursor. To end insertion, press [2nd] [INS] or a cursor key. |
| [2nd] | Next key press is a **2nd** operation (the gold-colored label printed to the left above a key). The cursor changes to an ▮ To cancel **2nd**, press [2nd] again. |
| [ALPHA] | Next key press is an **ALPHA** character (the light grey character to the right above a key). The cursor changes to an **A**. To cancel **ALPHA**, press [ALPHA] or a cursor key. |
| [2nd] [A-LOCK] | Sets **ALPHA-LOCK**; each subsequent key press results in an **ALPHA** character. The cursor changes to an **A**. To cancel **ALPHA-LOCK**, press [ALPHA]. |
| [X,T] | Allows you to enter an **X** in **FUNC** mode or a **T** in **PARAM** mode without first pressing [ALPHA]. |

# Setting Modes

**Modes control how numbers and graphs are displayed and interpreted by the calculator. Mode settings are retained by the Constant Memory feature when the TI-80 is turned off.**

## Checking MODE Settings

Press [MODE] to display the **MODE** screen. The current settings are highlighted. The settings are described on the following pages.

| | |
|---|---|
| NORMAL SCI | Numeric display format. |
| FLOAT 0123456789 | Number of decimal places. |
| RADIAN DEGREE | Unit of angle measure. |
| a⌴b/c b/c | Type of fraction display. |
| AUTOSIMP MANSIMP | Whether to simplify fractions. |
| FUNC PARAM | Type of graphing. |
| CONNECTED DOT | Whether to connect graph points. |
| SEQUENTIAL SIMUL | Whether to plot simultaneously. |

## Changing MODE Settings

To change the mode setting:

1. Press [▼] or [▲] to move the cursor to the line of the setting that you want to change. The setting that the cursor is on flashes.
2. Press [▶] or [◀] to move the cursor to the setting that you want.
3. Press [ENTER].

## Leaving the MODE Screen

To leave the **MODE** screen:

- Press the appropriate keys to go to another screen.
- Press [2nd] [QUIT] or [CLEAR] to return to the Home screen.

## Setting a Mode from a Program

You can set a mode from a program by entering the name of the mode as an instruction; for example, **FUNC** or **FLOAT**. From a blank line in the program editor (Chapter 10), press [MODE] to display a menu of the mode names, and then select the name. The name is copied to the cursor location.

# TI-80 Modes

**The TI-80 has eight mode settings. They control how numeric entries are interpreted, how answers are calculated or displayed, and how graphs appear in the display. Modes are set on the MODE screen (page 1-9).**

**NORMAL**
**SCI**

Notation formats affect only how an answer is displayed on the Home screen. Numeric answers can be displayed with up to 10 digits and a two-digit exponent. You can enter a number in any format.

**NORMAL** display format is the way in which we usually express decimal numbers, with digits to the left and right of the decimal point, as in **12345.67**.

**SCI** (scientific) notation expresses numbers in two parts. The significant digits can be displayed with one digit to the left of the decimal point. The appropriate power of 10 displays to the right of E, as in **1.234567E4**.

**Note:** If you select normal display format, but the answer cannot be displayed in 10 digits or the absolute value is less than .001, the TI-80 displays the answer in scientific notation.

**FLOAT**
**Fixed Decimal**

Decimal settings affect only how an answer is displayed on the Home screen. You can enter a number in any format. The decimal settings apply to both notation formats.

**FLOAT** (floating) decimal setting displays up to 10 digits, plus the sign and decimal.

The fixed decimal setting lets you select the number of digits (**0** to **9**) to be displayed to the right of the decimal point. The displayed value is rounded based on the number of digits you selected. The actual value is stored and used in calculations. Place the cursor on the number of decimal digits you want, and press [ENTER].

**Note:** In the program editor, the format for fixed decimal settings is **FIX** $n$. Enter $n$ as an integer from **0** to **9**. The mode is changed to fixed decimal when the program is executed.

| | |
|---|---|
| **RADIAN**<br>**DEGREE** | The angle mode controls:<br>• How the calculator interprets angle arguments in **SIN**, **COS**, **TAN**, and polar-to-rectangular conversions.<br>• How the calculator returns angle answers to **SIN**$^{-1}$, **COS** $^{-1}$, **TAN**$^{-1}$, and rectangular-to-polar conversions.<br><br>**RADIAN** mode interprets angle arguments as radians and returns angle answers in radians.<br><br>**DEGREE** mode interprets angle arguments as degrees and returns angle answers in degrees. |
| **a␣b/c**<br>**b/c** | **a␣b/c** displays fraction results as mixed numbers; for example, the result of **1/3 + 4/3** is displayed as **1␣2/3**.<br><br>**b/c** displays fraction results as simple fractions; for example, the result of **1/3 + 4/3** is displayed as **5/3**. |
| **AUTOSIMP**<br>**MANSIMP** | **AUTOSIMP** automatically simplifies fraction results to their lowest terms before displaying them; for example, the result of **2/6 + 2/6** is displayed as **2/3**.<br><br>**MANSIMP** displays fraction results without automatic simplification; for example, the result of **2/6 + 2/6** is displayed as **4/6**. |
| **FUNC**<br>**PARAM** | **FUNC** (function) graphing plots functions where **Y** is expressed in terms of **X** (Chapter 4).<br><br>**PARAM** (parametric) graphing plots relations where **X** and **Y** are each expressed in terms of **T** (Chapter 5). |
| **CONNECTED**<br>**DOT** | **CONNECTED** draws line segments between the calculated points of the selected functions.<br><br>**DOT** plots only the calculated points of the selected functions. |
| **SEQUENTIAL**<br>**SIMUL** | **SEQUENTIAL** graphing evaluates and plots one function completely before the next function is evaluated and plotted.<br><br>**SIMUL** (simultaneous) graphing evaluates and plots all selected functions for a single value of **X**, one at a time. In Parametric mode, **X** and **Y** are selected pairs. They are evaluated and plotted a **T** value at a time. |

# Variable Names

**On the TI-80 you can enter, name, and use several types of data: numeric values (including fractions), lists, functions, and statistical plots.**

## Variables and Defined Items

The TI-80 uses both user-assigned and pre-assigned names for variables and other items saved in memory.

| Variable Type | Names |
|---|---|
| Numeric values (including fractions) | **A**, **B**, ..., **Z**, θ (single character only) |
| Lists | **L1**, **L2**, **L3**, **L4**, **L5**, **L6** (on the keyboard) |
| Functions | **Y1**, **Y2**, **Y3**, **Y4** (on the **Y=** editor in **FUNC** mode) |
| Parametric equations | **X1T/Y1T**, **X2T/Y2T**, **X3T/Y3T** (on the **Y=** editor in **PARAM** mode) |
| Statistical plots | **PLOT1**, **PLOT2**, **PLOT3** (on the **STAT PLOT** menu) |
| System variables | **XMIN**, **XMAX**, and others (on various menus) |

Programs also have user-defined names and share memory with variables. Program names can be up to seven characters long. Programs are entered and edited from the program editor (Chapter 10).

You can store to lists (Chapter 8), system variables such as **XMAX** (Chapter 4) or **TBLMIN** (Chapter 6), and all **Y=** functions (Chapters 4 and 5) from the Home screen or from a program. You can store to lists (Chapters 8 and 9) and functions (Chapters 4 and 5) from editors. You can also store to a list element (Chapter 8).

For more information about system variables, see Appendix A.

# Storing and Recalling Variable Values

**Values are stored to and recalled from memory using variable names. When an expression containing the name of a variable is evaluated, the value of the variable at that time is used.**

## Storing Values in a Variable

You can store a value to a variable from the Home screen or a program using the [STO▸] key. Begin on a blank line.

1. Enter the value that you want to store (this can be an expression).
2. Press [STO▸]. The symbol → is copied to the cursor location.
3. Press [ALPHA] and then the single letter of the variable to which you want to store the value.
4. Press [ENTER]. If you entered an expression, it is evaluated. The value is stored in the variable.

```
5+4²→J
                21
```

## Displaying a Variable Value

To display the value of a variable, enter the variable name on a blank line on the Home screen, and then press [ENTER]. You can enter the name of the variable in one of the following ways:

- Press [ALPHA] and the letter of the variable (for user-defined variables).
- Press [2nd] and the name of the list.
- Press [VARS] and select the type and name of the variable (for system variables).
- Press [2nd] [Y-VARS] and select the type and name of the function.

## Using a Variable in an Expression

To use the current value of a variable in an expression, just enter the variable name in the expression.

```
2J
                42
L1-5
          {15 25 35}
```

# Last Entry

**When you press [ENTER] on the Home screen to evaluate an expression or execute an instruction, the expression or instruction is stored in an area called Last Entry, which you can recall. When you turn the TI-80 off, Last Entry is retained in memory.**

## Using Last Entry

You can recall Last Entry and edit it from the Home screen. Press [2nd] [ENTRY]. The current line is cleared, and the Last Entry is copied to the line. The cursor is positioned at the end of the entry. Because the TI-80 updates the Last Entry storage area only when you press [ENTER], you can recall the previous entry even if you have begun entering the next expression. However, when you recall Last Entry, it replaces what you have typed.

5 [+] 7
[ENTER]
[2nd] [ENTRY]

```
5+7
                    12
5+7
```

## Displaying a Previous Entry

The TI-80 keeps previous entries (up to a total of 80 bytes) in Last Entry. You can display and edit those entries by continuing to press [2nd] [ENTRY]. Last Entry displays previous entries in a loop, beginning with newest entry and moving to the oldest entry. Once the oldest item is displayed, [2nd] [ENTRY] displays the newest item again.

1 [STO▸] [ALPHA] A
[ENTER]
2 [STO▸] [ALPHA] B
[ENTER]
3 [STO▸] [ALPHA] C
[ENTER]
[2nd] [ENTRY]

```
1->A
                    1
2->B
                    2
3->C
                    3
3->C
```

When you press [2nd] [ENTRY] again, the previous item replaces the item on the current line.

[2nd] [ENTRY]

```
1->A
                    1
2->B
                    2
3->C
                    3
2->B
```

### Re-executing the Previous Entry

To execute Last Entry, press [ENTER] on a blank line on the Home screen. The entry is executed, but it does not reappear.

0 [STO▸] [ALPHA] N
[ENTER]
[ALPHA] N [+] 1 [STO▸] [ALPHA] N
[ENTER]
[ENTER]
[ENTER]

```
0->N
                0
N+1->N
                1
                2
                3
```

### Multiple Entries on a Line

To enter more than one expression or instruction on a line, separate them with a colon (:). They are all stored together in Last Entry.

If the previous entry contained more than one expression or instruction, separated with a colon (page 1-6), they are all recalled. You can recall all entries on a line, edit any of them, and then execute all of them.

Using the equation $A=\pi r^2$, use trial and error to find the radius of a circle that covers 200 square centimeters. Use 8 as your first guess.

8 [STO▸] [ALPHA] R [2nd] [:]
[2nd] [π] [ALPHA] R [$x^2$]
[ENTER]
[2nd] [ENTRY]

```
8->R:πR²
      201.0619298
8->R:πR²
```

Now try this.

[2nd] [◂] 7 [2nd] [INS] .95
[ENTER]

```
8->R:πR²
      201.0619298
7.95->R:πR²
      198.5565097
```

Continue until the answer is as accurate as you want.

# Last Answer

**When an expression is evaluated successfully from the Home screen or from a program, the TI-80 stores the answer to a variable, ANS (Last Answer). ANS may be a decimal number, a fraction, or a list. When you turn the TI-80 off, the value in ANS is retained in memory.**

## Using Last Answer (ANS) in an Expression

You can use the variable **ANS** to represent the last answer in most places. When you press [2nd] [ANS], the variable name **ANS** is copied to the cursor location. When the expression is evaluated, the TI-80 uses the value of **ANS** in the calculation.

Calculate the area of a garden plot 1.7 meters by 4.2 meters. Then calculate the yield per square meter if the plot produces a total of 147 tomatoes.

1.7 [×] 4.2
[ENTER]
147 [÷] [2nd] [ANS]
[ENTER]

```
1.7×4.2
                  7.14
147/Ans
       20.58823529
```

## Continuing an Expression

You can use the value in **ANS** as the first entry in the next expression without entering the value again or pressing [2nd] [ANS]. On a blank line on the Home screen, enter the function. The TI-80 "types" the variable name **ANS** followed by the function.

5 [÷] 2
[ENTER]
[×] 9.9
[ENTER]

```
5/2
                  2.5
Ans×9.9
                 24.75
```

## Storing Answers

To store an answer, store **ANS** to a variable before you evaluate another expression.

Calculate the area of a circle of radius 5 meters. Then calculate the volume of a cylinder of radius 5 meters and height 3.3 meters. Store the result in the variable **V**.

[2nd] [π] 5 [$x^2$]
[ENTER]
[×] 3.3
[ENTER]
[STO▸] [ALPHA] V
[ENTER]

```
π5²
       78.53981634
Ans×3.3
       259.1813939
Ans->V
       259.1813939
```

**To leave the keyboard uncluttered, the TI-80 uses full-screen menus to give you access to many additional operations. The use of specific menus is described in the appropriate chapters.**

### Moving from One Menu to Another

Some menu keys, such as [MATH], display more than one menu. The names of the menus appear on the top line. The current menu is highlighted and the items in that menu are displayed.

Press [▶] or [◀] to move the cursor to a different menu.

### Selecting an Item from a Menu

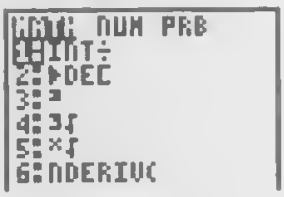
The number of the current item is highlighted. If there are more than seven items on the menu, a ↓ appears on the last line in place of the : (colon) between the menu number and name. Menu items, such as **VARS WINDOW**, that end in ... (ellipsis marks) display another menu.

There are two methods of selecting from a menu.

- Press the number of the item you want to select.
- Press [▼] and [▲] to move the cursor to the item you want to select, and then press [ENTER].

Calculate $\sqrt[3]{27}$.

1. Press [MATH] to display the **MATH** menu.




2. To select $\sqrt[3]{}$, you may either press **4** or press [▼] [▼] [▼] [ENTER].
3. Enter **27**, and then press [ENTER] to evaluate the expression.

### Leaving a Menu without Making a Selection

There are several ways to leave a menu without making a selection from the menu.

- To return to the Home screen, press [2nd] [QUIT].
- To return to the screen where you were, press [CLEAR].
- To display a different menu, press the appropriate key, such as [ZOOM].
- To select another screen, press the appropriate key, such as [WINDOW].

**You may want to use the names of system variables (such as XMIN) and functions (such as $Y_1$) in an expression. You may also want to store values directly to those variables. Use the VARS or Y-VARS menus to access the names.**

## VARS Menu

The **VARS** menu displays the names of window variables such as **XMIN** and **TSTEP**, statistics variables such as $\bar{x}$ and $Q_1$, and table variables such as **TBLMIN**.

Press [VARS] to display the **VARS** menu. Some of the items display more than one menu of variable names.

| VARS | |
|---|---|
| 1:WINDOW... | Names of **X**, **Y**, and **T** variables |
| 2:STATISTICS... | **X/Y**, Σ, **EQ**, and **BOX** variables |
| 3:TABLE... | **TBLMIN** and **ΔTBL** variables |
| 4:SIMPFACTOR... | Factor last used by **▶SIMP** function |

## Y-VARS Menu

The **Y-VARS** menus display the names of functions and the instructions to select or deselect functions from a program or the Home screen.

Press [2nd] [Y-VARS] to display the **Y-VARS** menu. Then press [▶] or [◀] to select the type of variable you want.

| | |
|---|---|
| **Y** | Displays a menu of names of $Yn$ functions. |
| **XT/YT** | Displays a menu of names of $XnT/YnT$ equations. |
| **ON/OFF** | Lets you select/deselect functions. |

## Copying a Name from a VARS or Y-VARS Menu

To copy a variable name from a **VARS** or **Y-VARS** menu:

1. Press [VARS] or [2nd] [Y-VARS]. The **VARS** or **Y-VARS** menu is displayed.
2. Select the type of variable you want.
3. Press [ENTER] to select the name you want from the menu. It is copied to the cursor location.

# EOS (Equation Operating System)

**The Equation Operating System (EOS™) defines the order of operations for the calculator—that is, the order in which the TI-80 evaluates functions in expressions. EOS lets you enter numbers and functions in a simple, straightforward sequence.**

## Order of Evaluation

A function returns a value. EOS evaluates the functions in an expression in the following order:

| | |
|---|---|
| 1 | Functions that are entered after the argument, such as **A²**, **2⁻¹**, **22!**, **45°**, **2π⌜**, and **▸SIMP**. |
| 2 | Powers and roots, such as **2^5** or **5 ˣ√32**. |
| 3 | Implied multiplication where the second argument is a number, variable name, or list, or begins with an open parenthesis, such as **4A**, **(A+B)4**, or **4(A+B)**. |
| 4 | Single-argument functions that precede the argument, such as **⁻A**, **√63**, **SIN B**, or **LOG 3**. |
| 5 | Implied multiplication where the second argument is a multi-argument function or a single-argument function that precedes the argument, such as **2NDERIV(A²,A,)** or **ASIN 2**. |
| 6 | Permutations (**nPr**) and combinations (**nCr**). |
| 7 | Multiplication and division (including **INT÷**). |
| 8 | Addition and subtraction. |
| 9 | Test functions, such as **>** or **≤**. |
| 10 | Conversion functions: **▸FRAC**, **▸DEC**, **▸a⌴b/c**, and **▸b/c**. |

Within a priority group, EOS evaluates functions from left to right. However, two or more single-argument functions that precede the same argument are evaluated from right to left. For example, **SIN FPART LN 8** is evaluated as **SIN(FPART(LN 8))**.

Calculations within a pair of parentheses are evaluated first. Multi-argument functions, such as **NDERIV(A²,A,6)**, are evaluated as they are encountered.

The conversion functions **▸FRAC**, **▸DEC**, **▸a⌴b/c**, and **▸b/c** can only be used at the end of a command line, with one exception: they can be followed by a store instruction.

**Implied Multiplication**

The TI-80 recognizes implied multiplication. For example, it understands **2π**, **4SIN 45**, **5(1+2)**, and **(2×5)7** as implied multiplication.

**Parentheses**

All calculations inside a pair of parentheses are completed first. For example, in the expression **4(1+2)**, EOS first evaluates the portion inside the parentheses, **1+2**, and then multiplies the answer, **3**, by **4**.

You can omit any right (closing) parenthesis at the end of an expression. All "open" parenthetical elements are closed automatically at the end of an expression and preceding the → (store) or display-conversion instructions.

**Note:** Parentheses are also used to enclose the arguments for certain functions, for example, **NDERIV(A$^2$,A,6)**. In these cases, parentheses do not indicate implied multiplication.

**Negation**

To enter a negative number, use the negation function. Press [(-)], and then enter the number. On the TI-80, negation is in the fourth group in the EOS hierarchy. Functions in the first group, such as squaring, are evaluated before negation.

For example, **-X$^2$** is a negative number (or 0); **-9$^2$** is **-81**. Use parentheses to square a negative number: **(-9)$^2$**.

```
-2²                  -4
(-2)²                 4
-A²(-A)²            -16
```

**Note:** Use the [−] key for subtraction and the [(-)] key for negation. If you press [−] to enter a negative number, as in **9** [×] [−] **7**, or if you press [(-)] to indicate subtraction, as in **9** [(-)] **7**, it is an error. If you press [ALPHA] **A** [(-)] [ALPHA] **B**, it is interpreted as implied multiplication (**A×-B**).

# Error Conditions

**The TI-80 detects any errors at the time it evaluates an expression, executes an instruction, plots a graph, or stores a value. Calculations stop and an error message with a menu is displayed immediately. Error codes and conditions are described in detail in Appendix B.**

## Diagnosing an Error

If the TI-80 detects an error, it displays the error screen.


The top line indicates the general type of error, such as **SYNTAX** or **DOMAIN**. For additional information about each error message, see Appendix B.

- If you select **GOTO**, the cursor is displayed at the location where the error was detected.

  **Note:** If a syntax error was detected in the contents of a **Y=** function during program execution, **GOTO** returns the user to the **Y=** editor, not to the program.

- If you select **QUIT**, or press [2nd] [QUIT] or [CLEAR], you return to the Home screen.

## Correcting an Error

To correct an error:

1. Note the type of the error.
2. Select **GOTO**, if that option is available, and look at the expression for syntax errors, especially at and before the cursor location.
3. If the error in the expression is not readily apparent, turn to Appendix B and read the information about the error message.
4. Correct the expression.

# Chapter 2: Math, Angle, and Test Operations

**This chapter describes the math, angle, and relational operations available on the TI-80. The most commonly used functions are accessed from the keyboard; others are accessed through menus.**

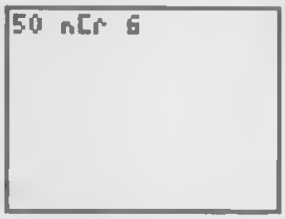
**Chapter Contents**

# Getting Started: Lottery Chances

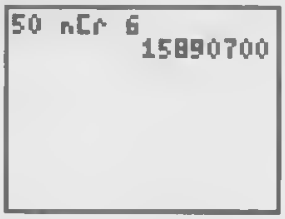
**Getting Started is a fast-paced introduction. Read the chapter for details.**

**Suppose you want to enter a lottery where six numbers will be drawn from 50. To win, you must pick all six numbers (in any order). What Is the probability of winning If you buy one ticket? What is the probability of winning if you buy five tickets?**

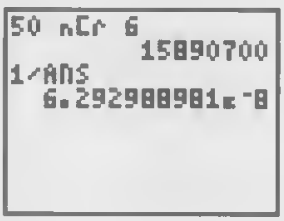
1. Determine the number of combinations possible. On the Home screen, press **50** to enter the total number of items. Press [MATH] [◀] to display the **MATH PRB** menu. Press **3** to select **nCr**. Press **6** to enter the number of items selected.



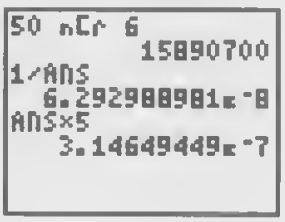

2. Press [ENTER] to evaluate the expression. This is the total number of possible combinations of 6 numbers drawn from a set of 50 numbers. With one ticket, you have one chance in 15,890,700 of winning.




3. To calculate the probability of winning with one ticket, press **1** [÷] [2nd] [ANS] [ENTER]. The answer is too large to display in fixed notation; therefore, it is shown in scientific notation. 0.00000006292988981 is the decimal equivalent.




4. To calculate the probability of winning with five tickets, press [×] **5** [ENTER]. Again, the answer is too small to display in fixed notation. The decimal equivalent is 0.000000314649449.



**This page contains some general information you should know about the TI-80 functions described in Chapter 1.**

### Using Lists with Functions

Functions that are valid for lists return a list calculated on an element-by-element basis. If two lists are used in the same expression, they must be the same length. For more information about lists, see Chapter 8.

```
{1,2}+{3,4}
                {4 6}
2×{1,2,3}
              {2 4 6}
```

### Using Fractions with Functions

Some math functions (+, –, ×, /, $x^2$, ▶**b/c**, ▶**a⌴b/c**, ▶**DEC**) accept fractions as input values. All other functions convert fractions to decimals before operating on them. For more information about fractions, see Chapter 3.

```
1/2+1/3
                  5/6
SIN 1/2
        .4794255386
√4/9
        .6666666667
```

# Keyboard Math Operations

**The most commonly used math functions are on the keyboard. The keyboard math operations can be used with decimal numbers, fractions (except as noted), expressions, and lists.**

**+ (Add)**
**− (Subtract)**
**× (Multiply)**
**/ (Divide)**

The basic arithmetic functions are: addition [+], subtraction [−], multiplication [×], and division [÷]. Each argument for these functions can be a list.

*valueA*+*valueB*, *valueA*−*valueB*, *valueA*×*valueB*, *valueA*/*valueB*

```
7+5-3×2
                    6
{1,2,3}+2
              {3 4 5}
```

**Trig Functions**

The trigonometric functions are interpreted according to the current **RADIAN/DEGREE** mode setting. (Refer to page 1-9 for instructions on changing the mode setting.) For example, **SIN 30** in **RADIAN** mode returns **-.9880316241**; in **DEGREE** mode, it returns **.5**. Each argument for the trigonometric functions may be a list.

**SIN** *value*, **COS** *value*, **TAN** *value*

**SIN$^{-1}$**, **COS$^{-1}$**, and **TAN$^{-1}$** are the inverse trig functions (arcsine, arccosine, and arctangent).

**SIN$^{-1}$** *value*, **COS$^{-1}$** *value*, **TAN$^{-1}$** *value*

**RADIAN Mode**

```
COS (π/2)
                    0
COS⁻¹ ANS
          1.570796327
(SIN 0)²+(COS 0)
²
                    1
```

**$^{-1}$ (Inverse)**

$^{-1}$ (inverse, [$x^{-1}$]) may be used with numbers, expressions, or lists. The multiplicative inverse is the equivalent of the reciprocal, 1/x.

*value*$^{-1}$

```
5⁻¹
                   .2
{4,2}⁻¹
             {.25 .5}
```

**^ (Power)**
**2 (Square)**
**√ (Square Root)**

^ (power, [^]), 2 (square, [x²]), and √ (square root, [2nd] [√]) may be used with decimal numbers, fractions, expressions, or lists. When used with a fraction, √ (square root, [2nd] [√]) returns a decimal number.

*value*^*power*, *value*2, √*value*

**Note:** Raising a negative number to a noninteger power can result in a complex number, which returns an error.

```
25²
                  625
√(ANS)
                    25
{1,2,3}^3
              {1 8 27}
```

**LOG**
**10^**
**LN**

These functions find the logarithm [LOG], power of ten [2nd] [10^x], and natural log [LN] of the specified value or list of values.

**LOG** *value*, **10^***power*, **LN** *value*

```
LOG 66
        1.819543936
10^(ANS)
                  66
LN {1,2}
 {0 .6931471806}
```

**e^**

**e^** ( [2nd] [e^x]) returns the constant **e** raised to a power or list of powers. **e^1** returns the value of the constant **e**.

**e^***power*

```
e^5
        148.4131591
e^1
        2.718281828
```

**⁻ (Negation)**

⁻ (negation, [(-)]) returns the negative of a number, expression, or list. The narrow negation symbol (⁻) distinguishes negation from the subtraction or minus (–).

⁻*value*

EOS rules (Chapter 1) determine when negation is evaluated. For example, $^{-}A^2$ returns a negative number because squaring is evaluated before negation. Use parentheses to square a negated number, $(^{-}A)^2$.

```
{⁻A²,(⁻A)²,⁻2²,(
⁻2)²}
      {⁻4 4 ⁻4 4}
```

**ABS**

**ABS** (absolute value, [2nd] [ABS]) returns the absolute value of a number, expression, or list.

**ABS** *value*

```
ABS ⁻256
                256
ABS {1.25,⁻5.67}
     {1.25 5.67}
```

**π (Pi)**

Pi ([2nd] [π]) is stored as a constant in the TI-80. Press [2nd] [π] to copy the symbol π to the cursor location. The number **3.141592654** is displayed for π, but 3.1415926535898 is used internally in calculations.

```
2π
      6.283185307
πr²
      78.53981634
```

**To display the MATH MATH menu, press [MATH]. When you select a menu item, the name is copied to the cursor location. Functions that are valid for lists return a list calculated on an element-by-element basis.**

**MATH MATH Menu**

| MATH NUM PRB | |
|---|---|
| 1: INT÷ | Displays quotient and remainder |
| 2: ▶DEC | Displays answer in decimal form |
| 3: $^3$ | Cube |
| 4: $\sqrt[3]{}$ | Cube root |
| 5: $\sqrt[x]{}$ | $n^{th}$ root |
| 6: NDERIV( | Numerical derivative |

**INT÷**

On the Home screen or from a program, **INT÷** (integer divide, **MATH MATH**, item 1) returns the quotient (or quotient and remainder) resulting from the division of two integers. Each argument can be a list.

*integerA* **INT÷** *integerB*

When **INT÷** is executed on the Home screen, it displays the symbols **Q=** for the quotient and **R=** for the remainder.

**Note:** If **INT÷** is embedded in an expression, **Q=** and **R=** may not be displayed.

When used with lists, **INT÷** returns a list of quotients only.

If the result of **INT÷** is used in subsequent calculations, the remainder is dropped, and only the quotient is used.

The quotient from **INT÷** is returned to **ANS**.

```
9 INT÷ 2
 Q=4
 R=1
9 INT÷ 2+3
                7
5 INT÷ {1,2,3}
          {5 2 1}
```

**▶DEC**

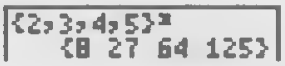
**▶DEC** (convert to decimal, **MATH MATH**, item 2) displays an answer in decimal form. **▶DEC** can only be used after a *value* and at the end of an *expression*. *value* can be a list.

*value*▶**DEC**
*expression*▶**DEC**


**$^3$ (Cube)**

$^3$ (cube, **MATH MATH**, item 3) returns the cube of a number, expression, or list.

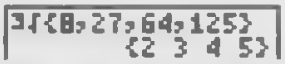
*value*$^3$




**$\sqrt[3]{}$ (Cube Root)**

$\sqrt[3]{}$ (cube root, **MATH MATH**, item 4) returns the cube root of a number, expression, or list.

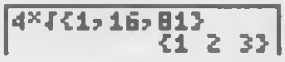
$\sqrt[3]{}$*value*




**$\sqrt[x]{}$ (Root)**

$\sqrt[x]{}$ (root, **MATH MATH**, item 5) returns the $n^{th}$ real root of a number, expression, or list.

$n^{th}root\sqrt[x]{}value$



**NDERIV(**

**NDERIV(** (numerical derivative, **MATH MATH**, item 6) returns an approximate derivative of an expression with respect to a specified variable, given the value at which to calculate the derivative, and ε (optional; if none is specified, 1ε⁻3 is used).

**NDERIV(***expression*,*variable*,*value*) or
**NDERIV(***expression*,*variable*,*value*,ε)

**NDERIV(** uses the symmetric difference quotient method (as shown in the formula below), which approximates the numerical derivative value as the slope of the secant line through the points:

$$f'(x) = \frac{f(X+\varepsilon)-f(X-\varepsilon)}{2\varepsilon}$$

As ε gets smaller, the approximation usually becomes more accurate.

```
nDERIV(A^3,A,5)
          75.000001
nDERIV(A^3,A,5,.
0001)
                  75
```

Because of the method used, **NDERIV(** can return a false derivative value at a nondifferentiable point.

**To display the MATH NUM menu, press MATH ▸. When you select a menu item, the name is copied to the cursor location. Functions that are valid for lists return a list calculated on an element-by-element basis.**

**MATH NUM Menu**

```
MATH NUM  PRB
1:ROUND(          Round
2:IPART           Integer part
3:FPART           Fractional part
4:INT             Greatest integer
5:MIN(            Minimum value
6:MAX(            Maximum value
7:REMAINDER(      Remainder of a division result
```

**ROUND(**

**ROUND(** (**MATH NUM**, item 1) returns a number, expression, or list rounded to a specified number of decimals (≤ 9). If the number of decimals is omitted, the number is rounded to the digits that are displayed, a maximum of 10 digits.

**ROUND(***value*,*#decimals*)

```
ROUND(π,4)
                3.1416
ROUND(π,{1,2,3}
)
{3.1 3.14 3.142}
```

**ROUND(***value*)

```
123456789012->C
    1.23456789E11
C-ROUND(C)
                   12
123456789012-123
456789000
                   12
```

**IPART**
**FPART**

**IPART** (integer part, **MATH NUM**, item 2) returns the integer part or parts of a number, expression, or list. **FPART** (fractional part, **MATH NUM**, item 3) returns the fractional part or parts of a number, expression, or list.

**IPART** *value*

```
IPART -23.45
                  -23
IPART {5.2,-2.6}
                {5 -2}
```

**FPART** *value*

```
FPART -23.45
                 -.45
FPART {5.2,-2.6}
              {.2 -.6}
```

**INT**

**INT** (greatest integer, **MATH NUM**, item 4) returns the largest integer less than or equal to a number, expression, or list. The value is the same as **IPART** for nonnegative numbers and negative integers, but one integer less than **IPART** for negative noninteger numbers.

**INT** *value*

```
INT -23.45
                  -24
INT 23.45
                   23
```

**MIN(**
**MAX(**

**MIN(** (minimum value, **MATH NUM**, item 5) returns the smaller of two values or the smallest element in a list. If two lists are compared, it returns a list of the smaller of each pair of elements. If a list and a value are compared, it compares each element in the list with the value.

**MAX(** (maximum value, **MATH NUM**, item 6) returns the larger of two values or the largest element in a list. If two lists are compared, it returns a list of the larger of each pair of elements. If a list and a value are compared, it compares each element in the list with the value.

**MIN(***valueA*,*valueB*)
**MIN(***list*)
**MIN(***listA*,*listB*)
**MIN(***value*,*list*)
**MIN(***list*,*value*)

**MAX(***valueA*,*valueB*)
**MAX(***list*)
**MAX(***listA*,*listB*)
**MAX(***value*,*list*)
**MAX(***list*,*value*)

```
MIN(3,2+2)
                  3
MIN({3,4,5},4)
              {3 4 4}
MAX({4,5,6})
                  6
```

Note: **MIN(** and **MAX(** are also available on the **LIST MATH** menu.

**REMAINDER(**

**REMAINDER(** (**MATH NUM**, item 7) returns the remainder resulting from the division of two integers, each of which can be a list. (See **INT÷**, page 2-7.)

**REMAINDER(***valueA*,*valueB*)
**REMAINDER(***value*,*list*)
**REMAINDER(***listA*,*listB*)
**REMAINDER(***list*,*value*)

If a list is used as one or both arguments, the result is a list of remainders.

```
REMAINDER(10,4)
                  2
REMAINDER({15,16
,17},5)
              {0 1 2}
```

**To display the MATH PRB menu, press MATH ◀. When you select a menu item, the name is copied to the cursor location. Functions that are valid for lists return a list calculated on an element-by-element basis.**

**MATH PRB Menu**

| MATH NUM PRB | |
|---|---|
| 1:RAND | Random number generator |
| 2:nPr | Number of permutations |
| 3:nCr | Number of combinations |
| 4:! | Factorial |
| 5:RANDINT( | Random integer generator |

**RAND**

**RAND** (random number, **MATH PRB**, item 1) generates and returns a random number greater than 0 and less than 1 (as in the first example below). A random number is generated from a seed value. To control a random number sequence, first store an integer seed value in **RAND**. In the second example below, **1** is stored to **RAND** so that the TI-80 uses 1 as the seed value for generating random numbers.

```
RAND
          .59566052
          .049599836
          .876572691
          .077548484
          .525931944
```

```
1->RAND
                    1
RAND
          .601079504
          .838032464
          .483957029
          .007661969
```

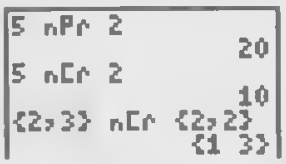
**Note:** When you reset the TI-80, **RAND** is set to the factory seed value, which is 0.

**nPr**
**nCr**

**nPr** (number of permutations, **MATH PRB**, item 2) returns the number of **permutations** of *items* taken *number* at a time. *items* and *number* must be nonnegative integers. Both *items* and *number* can be lists.

**nCr** (number of combinations, **MATH PRB**, item 3) returns the number of **combinations** of *items* taken *number* at a time. *items* and *number* must be nonnegative integers. Both *items* and *number* can be lists.

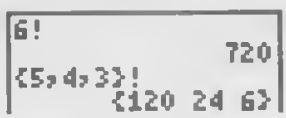
*items* **nPr** *number*
*items* **nCr** *number*




**! (Factorial)**

**!** (factorial, **MATH PRB**, item 4) returns the factorial of a positive integer or list of integers between 0 and 69.

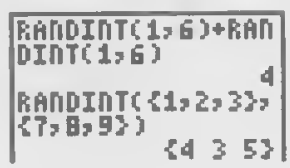
*value*!




**RANDINT(**

**RANDINT(** (random integer, **MATH PRB**, item 5) generates a random integer within a specified range. It requires two arguments: the lower and upper boundaries of the range (in any order). Both arguments must be integers. Both arguments can be negative. Both arguments can be lists.

**RANDINT(***lower*,*upper*)



**To display the ANGLE menu, press [2nd] [ANGLE]. The ANGLE menu displays angle indicators and instructions. When you select an item from the menu, the name is copied to the cursor location.**

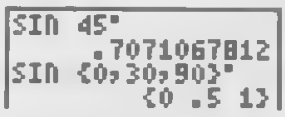
**ANGLE Menu**

| ANGLE | |
|---|---|
| 1: ° | Degree notation |
| 2: r | Radian notation |
| 3: R▶Pr( | Returns **R**, given **X** and **Y** |
| 4: R▶Pθ( | Returns θ, given **X** and **Y** |
| 5: P▶Rx( | Returns **X**, given **R** and θ |
| 6: P▶Ry( | Returns **Y**, given **R** and θ |

**°(Degree)**

° (degree, **ANGLE**, item 1) lets you designate an angle or list of angles as degrees, regardless of the current angle mode setting. In **RADIAN** mode, ° can also be used to convert degrees to radians.

*angle*°

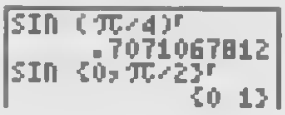



**RADIAN Mode**

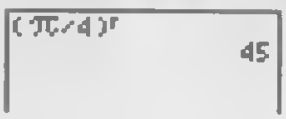

**r (Radians)**

r (radians, **ANGLE**, item 2) lets you designate an angle or list of angles as radians, regardless of the current angle mode setting. In **DEGREE** mode, r can also be used to convert radians to degrees.

*angle*r




**DEGREE Mode**



**R▸Pr(**
**R▸Pθ(**
**P▸Rx(**
**P▸Ry(**

**Note:** When converting from one coordinate system to the other, be sure that the angle mode setting, **DEGREE** or **RADIAN**, is appropriate for your angle measurements. (Press MODE to check the current setting.)

**R▸Pr(** (**ANGLE**, item 3) converts the given rectangular coordinates to polar coordinates and returns **R**.

**R▸Pθ(** (**ANGLE**, item 4) converts the given rectangular coordinates to polar coordinates and returns θ.

Both $X$ and $Y$ can be lists.

**R▸Pr**($X$,$Y$)
**R▸Pθ**($X$,$Y$)

**RADIAN Mode**

```
R▸Pr(-1,0)
                    1
R▸Pθ(-1,0)
          3.141592654
```

**P▸Rx(** (**ANGLE**, item 5) converts the given polar coordinates to rectangular coordinates and returns **X**.

**P▸Ry(** (**ANGLE**, item 6) converts the given polar coordinates to rectangular coordinates and returns **Y**.

Both $R$ and θ can be lists.

**P▸Rx**($R$,θ)
**P▸Ry**($R$,θ)

**RADIAN Mode**

```
P▸Rx(1,π)
                   -1
P▸Ry(1,π)
                    0
```

# TEST (Relational) Operations

**To display the TEST menu, press [2nd] [TEST]. When you select from the menu, the name is copied to the cursor location. These functions are valid for lists; they return a list calculated on an element-by-element basis.**

## TEST Menu

| TEST | True if: |
|---|---|
| 1:= | Equal to |
| 2:≠ | Not equal to |
| 3:> | Greater than |
| 4:≥ | Greater than or equal to |
| 5:< | Less than |
| 6:≤ | Less than or equal to |

=
≠
>
≥
<
≤

Relational operators compare *valueA* and *valueB* and return **1** if the test is true or **0** if the test is false. *valueA* and *valueB* can be numbers, expressions, or lists.

Relational operators are often used in programs to control program flow and in graphing to control the graph of a function over specific values.

*valueA*=*valueB*
*valueA*≠*valueB*
*valueA*>*valueB*
*valueA*≥*valueB*
*valueA*<*valueB*
*valueA*≤*valueB*

```
25=26
                    0
{1,2,3}<3
                {1 1 0}
{1,2,3}≠{3,2,1}
                {1 0 1}
```

## Using Tests

Relational operators are evaluated after mathematical functions according to EOS rules (Chapter 1).

- The expression **2+2=2+3** returns **0**. The TI-80 does the addition first because of EOS rules, and then it compares 4 to 5.
- The expression **2+(2=2)+3** returns **6**. The TI-80 first performs the relational test because it is in parentheses; then it adds 2, 1, and 3.

# Chapter 3: Fractions

**This chapter describes how to use the fraction operations on the TI-80.**

**Chapter Contents**

# Getting Started: Working with Fractions

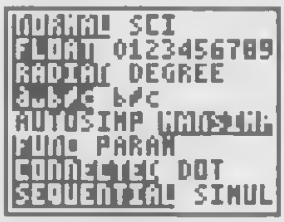
**Getting Started is a fast-paced introduction. Read the chapter for details.**

**Enter the expression 1 6/27 + 1 1/9. Evaluate the expression, simplify the result, and then use the conversion options on the FRACTION menu to convert the result.**

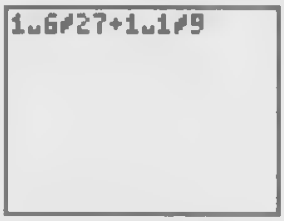
---

This example is performed in **MANSIMP** (manual simplification) mode. **MANSIMP** mode is especially useful for students when they are learning fraction concepts. When **MANSIMP** mode is selected, the **▶SIMP** function (from the **FRACTION** menu) can be used to simplify fractions, step-by-step.

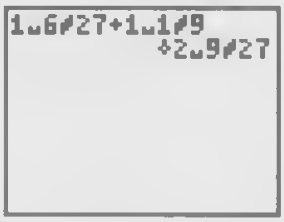
1. Select **MANSIMP** mode.



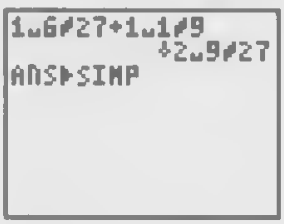

2. From the Home screen, press **1** [2nd] [UNIT⌴] **6** [2nd] [b/c] **27** [+] **1** [2nd] [UNIT⌴] **1** [2nd] [b/c] **9** to enter the mixed-fraction expression, **1 6/27 + 1 1/9**.




3. Press [ENTER] to evaluate the expression. The ↓ indicates that the fraction can be simplified.

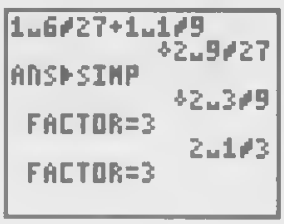



4. Press [FRAC] **1** to select **▶SIMP** (simplify). **ANS▶SIMP** is copied to the cursor location.



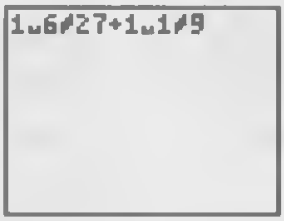

5. Press [ENTER] to simplify the fraction.

   In **MANSIMP** mode, the TI-80 uses the lowest common denominator for simplification. The simplification factor is displayed. The ↓ preceding the result indicates that the fraction can be simplified further. Continue pressing [ENTER] until ↓ is no longer displayed.

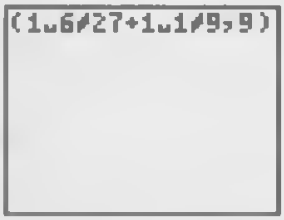


**The TI-80 uses the lowest common denominator for simplification. If you want to choose the simplification factor yourself, you can enter it as part of the expression.**

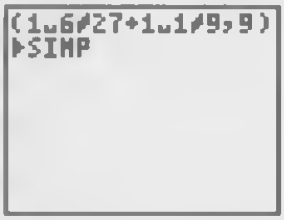
6. Press [CLEAR] to clear the screen. Reenter the expression, or press [2nd] [ENTRY] until you see the expression **1_6/27 + 1_1/9**.



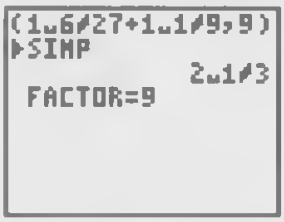

7. Press [2nd] [◄] [2nd] [INS] [(] [2nd] [►] [,] **9** [)]. This adds the simplification factor 9 and places the expression in parentheses.



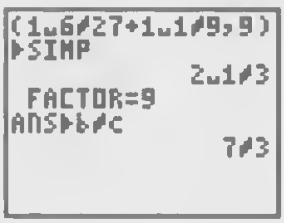

8. Press [FRAC] **1** to copy **►SIMP** to the cursor location.



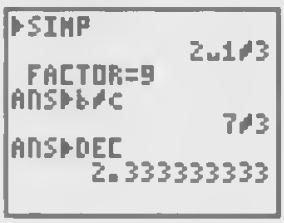

9. Press [ENTER] to simplify the fraction result. The simplification factor is displayed.




10. Press [2nd] [ANS] [FRAC] **2** [ENTER] to convert the mixed-fraction result to a simple fraction.




11. Press [2nd] [ANS] [FRAC] **5** [ENTER] to convert the fraction result to its decimal equivalent.



# Setting Modes for Fraction Results

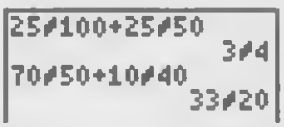
**From the MODE screen, you can select simplification and display format options for fraction results.**

### AUTOSIMP Mode with b/c and a⌴b/c Modes

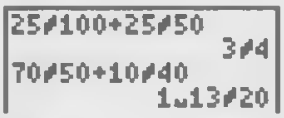
**AUTOSIMP** mode simplifies fractions automatically. Simplification takes place before the expression is evaluated. Then the result is simplified to its lowest terms. For example, **12/16** is simplified to **3/4** when you press ENTER.

There are two formats for displaying fractions results.

- **b/c** mode displays fraction results in simple-fraction (a fraction without a whole number) format; for example, **25/4**.



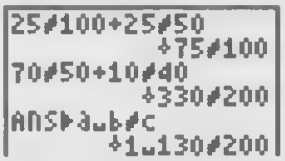

- **a⌴b/c** displays fraction results in mixed-fraction (a whole number with a fraction) format; for example **5 3/4**.



### MANSIMP Mode with a␣b/c Mode

**MANSIMP** mode lets you simplify fractions manually. **MANSIMP** was designed for teaching and learning fractions concepts. In **MANSIMP** mode, you can simplify fractions and the results of expressions using fractions, step-by-step.

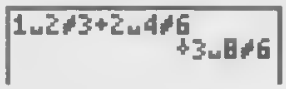
When a fraction result is not expressed in its lowest terms, a down arrow (↓) is displayed to remind you that you can simplify the result. Use ▶**SIMP** from the **FRACTION** menu to simplify the fraction. You can then use ▶**b/c** or ▶**a␣b/c** to change the display format of the fraction result.



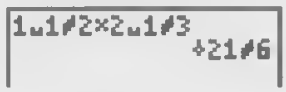

Typically, you use the **MANSIMP** simplification mode with the **a␣b/c** display format mode for teaching or learning fraction concepts. The display format of fraction results can vary when you are using **MANSIMP** and **a␣b/c**.

- When you simply enter a fraction and press ENTER, the format in which you entered the fraction is preserved.


- When you add or subtract using a mixed fraction, calculation takes place on the whole-number and fractional part of the mixed-fraction separately. The result is displayed as a mixed fraction.




- When you multiply or divide using a mixed fraction, the result is displayed as a simple fraction.



# Entering and Using Fractions in Calculations

**The TI-80 lets you enter fractions directly from the keyboard.**

## Entering Simple Fractions

A simple fraction is a fraction with no whole-number part; for example, **3/4** or **4/3**.

To enter a simple fraction:

1. Enter the numerator (up to six digits), and then press 2nd [b/c].
2. Enter the denominator (up to and including 1000).

For example, press **2** 2nd [b/c] **3** to enter **2/3**.

## Entering Mixed Fractions

A mixed fraction is fraction that has both a whole-number and a fractional part; for example **1 1/3**.

To enter a mixed fraction:

1. Enter the units (up to three digits), and then press 2nd [UNIT␣].
2. Enter the numerator (up to three digits), and then press 2nd [b/c].
3. Enter the denominator (up to and including 1000).

For example, press **5** 2nd [UNIT␣] **2** 2nd [b/c] **3** to enter **5 2/3**.

**In general, you can use fractions in expressions just as you would use other numbers. The results of the expressions, however, may or may not be fractions.**

## Using Fractions in Expressions

The absolute value of a fraction on the TI-80 cannot be ≥1000.

[+], [−], [×], [÷], [$x^{-1}$], [$x^2$], [(-)], and [2nd] [ABS] accept fraction entries and return fraction results. If the absolute value of a fraction result is ≥1000, or if the results of operations with these functions are not within the limits shown on page 3-6, the results are given in decimal form.

Other functions accept fraction entries, but convert them to decimal form before operating on them. The results are given in decimal form. For example, $\sqrt{}$ **4/9** returns **.6666666667**, not **2/3**.

If you use ▸**SIMP** with a fraction that has been converted to a decimal, an error occurs.

If an expression contains both a fraction and a decimal number, the result is displayed as a decimal number.

You can also enter fractions in a list, but the results are returned as decimal values.

```
1↵1/2+.25
                    1.75
1↵1/2-1
                     1/2
{1/4,2/4,3/4}
     {.25 .5 .75}
```

**To display the FRACTION menu, press FRAC. The menu items let you simplify and convert fractions. When you select a menu item, the name is copied to the cursor location.**

**FRACTION Menu**

| FRACTION | |
|---|---|
| 1: ▶SIMP | Simplifies the fraction. |
| 2: ▶b/c | Converts to a simple fraction. |
| 3: ▶a⌴b/c | Converts to a mixed fraction. |
| 4: ▶FRAC | Converts a decimal to a fraction based on mode. |
| 5: ▶DEC | Converts a fraction to a decimal. |

**Simplifying Fractions ▶SIMP**

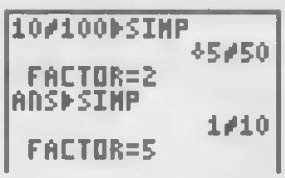
**▶SIMP** (simplify fraction, **FRACTION**, item 1) simplifies the specified fraction and displays it, along with the simplification factor.

**Note:** **▶SIMP** can only be used in **MANSIMP** mode.

You have two options for simplifying fractions.

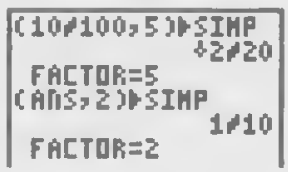
- You can let the calculator simplify the fraction, step-by-step, using the lowest common denominator (LCD), or

  *fraction*▶**SIMP**




- You can choose a factor (an integer) for simplifying the fraction.

  (*fraction*,*factor*)▶**SIMP**




Both simplification options update the variable **FACTOR**.

## Converting Simple and Mixed Fractions ▸b/c ▸a⌴b/c

**▸b/c** (convert to simple fraction, **FRACTION**, item 2) converts *value* to a simple fraction.

**▸a⌴b/c** (convert to mixed fraction, item 3) converts *value* to a mixed fraction.

*value*▸**b/c**
*value*▸**a⌴b/c**

```
4/3▸a⌴b/c
                1⌴1/3
4⌴3/2▸a⌴b/c
                5⌴1/2
Ans▸b/c
                11/2
```

Both ▸**b/c** and ▸**a⌴b/c** can be used only at the end of an expression. A → ([STO▸]) instruction, however, can follow them.

**Converting Decimals and Fractions**
**▶FRAC**
**▶DEC**

**▶FRAC** (convert to fraction, **FRACTION**, item 4) converts a *decimal* value to its fraction equivalent and displays it. The *decimal* may be a number, expression, or list.

In **MANSIMP** mode, **▶FRAC** first attempts to return a fraction in terms of 10ths, 100ths, or 1000ths. If this is not possible, **▶FRAC** converts the decimal to its fraction equivalent as it would in **AUTOSIMP** mode. If the value cannot be converted or if the denominator of the equivalent fraction is greater than 1000, the decimal equivalent is returned.

The form of the **▶FRAC** result depends on the current fraction display format. For example, **1.25 ▶FRAC** returns **1 1/4** if **a⌴b/c** is selected or **5/4** if **b/c** is selected.

If the *decimal* value for **▶FRAC** is a list, the list is displayed as fractions, but it is still stored internally in decimal form.

**▶DEC** (convert to decimal, **FRACTION**, item 5) converts a *fraction* value to its decimal form and displays it.

*decimal***▶FRAC**
*fraction***▶DEC**

**AUTOSIMP & a⌴b/c**

```
4/10+7/5
                  1⌴4/5
ANS▶DEC
                    1.8
ANS▶FRAC
                  1⌴4/5
```

**AUTOSIMP & b/c**

```
4/10+7/5
                    9/5
ANS▶DEC
                    1.8
ANS▶FRAC
                    9/5
```

**MANSIMP & a⌴b/c**

```
4/10+7/5
                 -18/10
ANS▶DEC
                    1.8
ANS▶FRAC
               -1⌴8/10
```

**MANSIMP & b/c**

```
4/10+7/5
                 -18/10
ANS▶DEC
                    1.8
ANS▶FRAC
                 -18/10
```

Both **▶FRAC** and **▶DEC** are valid only at the end of an expression. A → ([STO▶]) instruction, however, can follow them.

# Chapter 4: Function Graphing

**This chapter describes function graphing on the TI-80 in detail. It also lays the foundation for using the parametric graphing features described in Chapter 5.**

**Chapter Contents**

# Getting Started: Graphing a Circle

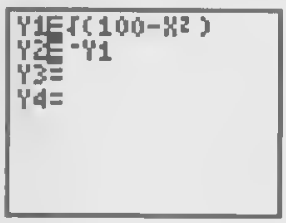
**Getting Started is a fast-paced introduction. Read the chapter for details.**

**Graph a circle of radius 10, centered on the origin in the standard viewing window. To graph a circle, you must enter separate formulas for the upper and lower portions of the circle. Then use ZSQUARE to adjust the display to make the functions appear as a circle.**

Make sure that your TI-80 is in **FUNC** mode and all **STAT PLOTS** are turned off.

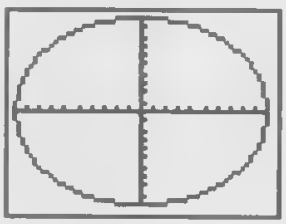
1. Press [Y=] to display the **Y=** edit screen. Press [2nd] [√] [(] **100** [−] [X,T] [$x^2$] [)] [ENTER] to enter the expression **Y1=√(100–X²)** to define the top half of the circle.

   The bottom half of the circle is defined by **Y2=-√(100–X²)**. However, you can also define one function in terms of another; so to define **Y2=-Y1**, press [(-)] [2nd] [Y-VARS] (to display the **Y=** variables menu) **1** (to select **Y1**).

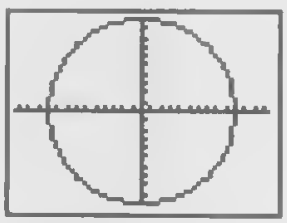



2. Press [ZOOM] **6** to select **ZSTANDARD**. This is a quick way to reset the Window variables to the standard values. It also graphs the functions; you do not need to press [GRAPH].

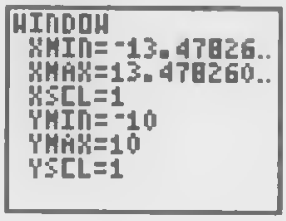
   Notice that the functions appear as an ellipse in the standard viewing window.



3. To adjust the display so each "dot" represents an equal width and height, press [ZOOM], and then **5** to select **ZSQUARE**. The functions are replotted and now appear as a circle on the display.



4. To see the **ZSQUARE** Window variables, press [WINDOW] and note the values for **XMIN**, **XMAX**, **YMIN**, and **YMAX**.




5. If you want to see the graph again, press [GRAPH].

# Defining a Graph

**To define a graph, you set the modes, enter and select the functions to graph, and define the viewing window and the graphing format. Once you have defined a graph, you can plot it, display it, and explore it.**

## Steps in Defining a Graph

There are six basic steps to defining a graph, although you may not need to take all of the steps each time you define a graph. The procedures are described in detail on the following pages.

1. Set the mode to **FUNC** graphing (Chapter 1).
2. Enter or edit a function in the **Y=** list (page 4-5).
3. Select the **Y=** function you want to graph (page 4-8).
4. Define the viewing window (page 4-9).
5. Set the graphing format (page 4-11).
6. Deselect **STAT PLOTS**, if appropriate (Chapter 9).

## Exploring a Graph

Once you have defined a graph, you can display it and use several tools on the TI-80 to explore the behaviour of the function or functions. These tools are described later in this chapter.

# Setting Graph Modes

**Pressing [MODE] displays the current mode settings, as described in Chapter 1. For function graphing, the graphing mode must be set to FUNC. Before you graph a function, check to make sure that the mode settings are appropriate.**

## Checking and Changing Graphing Modes

Press [MODE] to display the mode settings. The current settings are highlighted.

The TI-80 has two graphing modes.

- **FUNC** (function graphing)
- **PARAM** (parametric graphing)

To graph functions, you must select **FUNC** (function graphing).

The basics of graphing on the TI-80 are described in this chapter. Differences in parametric graphing are described in Chapter 5.

The mode settings can affect how functions are graphed.

- **RADIAN** or **DEGREE** mode may affect how some functions are interpreted.
- **CONNECTED** or **DOT** affects how the selected functions are plotted.
- **SEQUENTIAL** or **SIMUL** affects how functions are plotted if you have selected more than one function.

## Setting Modes from a Program

You can set the graphing mode and other modes from a program.

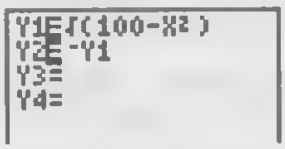
Begin on a blank line in the program editor. Press [MODE] to display the **MODE** screen. Press [▼] and [▲] to place the cursor on the mode that you want to select, and then press [ENTER]. The name of the mode is copied to the cursor location.

# Defining Functions in the Y= List

**Pressing [Y=] displays the Y= edit screen. This is where you enter the functions to graph. You can store up to four functions in memory at one time. You can graph one or more of these functions at a time.**

## Displaying the Functions in the Y= List

Press [Y=] to display the **Y=** edit screen. In the example below, the **Y1** and **Y2** functions are defined.




## Defining a New Function

To define a new function in the **Y=** list:

1. Press [Y=] to display the **Y=** edit screen.
2. Move the cursor to the function in the **Y=** list you want to define. If necessary, press [CLEAR] to erase a previously entered function.
3. Enter the expression to define the function.
   - You may use functions and variables (including lists) in the expression. If the expression evaluates to a value that is not a real number, that point is not plotted; an error does not occur.
   - The independent variable in the function is **X**. You may press [X,T], rather than pressing [ALPHA] [X], to enter the **X** variable. (**FUNC** mode defines the independent variable as **X**.)
   - The expression is stored as one of the four user-defined functions in the **Y=** list as you enter it.
4. When you complete the expression, press [ENTER] to move to the beginning of the next function.

**Note:** When you enter a function, it is automatically selected for graphing in the **Y=** list. This is indicated by the highlighted equal sign. For details on selecting and deselecting functions, see page 4-8.

### Editing a Function

To edit a function in the **Y=** list:

1. Press [Y=] to display the **Y=** list and move the cursor to the function you want to change.
2. Make the changes. You can also press [CLEAR] to erase the expression, and then enter a new expression.

The expression is stored in the **Y=** list and selected (turned on) as you edit it.

### Clearing a Function

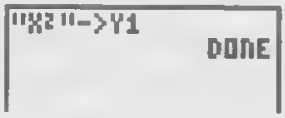
To clear or erase a function on the **Y=** edit screen, position the cursor anywhere on the function, and then press [CLEAR].

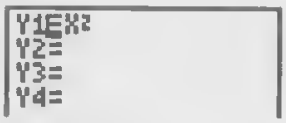
### Defining Functions from the Home Screen or a Program

To define a function from the Home screen or from a program, begin on a blank line.

1. Press [ALPHA] ["], enter the expression, and then press [ALPHA] ["] again.
2. Press [STO▸].
3. Press [2nd] [Y-VARS], and then select the name of the function from the **Y** menu. The name is copied to the cursor location.
4. Press [ENTER] to complete the instruction.

"*expression*"→**Y***n*




When the instruction is executed, the TI-80 stores the expression in the **Y=** list, selects (turns on) the function, and displays the message **DONE**.

### Leaving the Y= Edit Screen

To leave the **Y=** edit screen:

- Select another screen by pressing the appropriate key, such as [GRAPH] or [WINDOW].
- Press [2nd] [QUIT] to return to the Home screen.

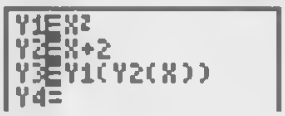
# Evaluating Y= Functions in Expressions

**You can the calculate the value of a Y= function at a specified value of X.**

## Entering the Functions in the Y= List

To display the **Y=** list, press [Y=]. Enter these functions for Y1, Y2, and Y3: **X**$^2$, **X+2**, and **Y1(Y2(X))**.

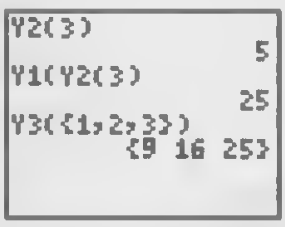
[X,T] [$x^2$] [ENTER]
[X,T] [+] **2** [ENTER]
[2nd] [Y-VARS] **1** [(] [2nd] [Y-VARS] **2**
[(] [X,T] [)] [)]




## Evaluating Functions

To evaluate the functions, first specify the value of X. Note that X may be a list.

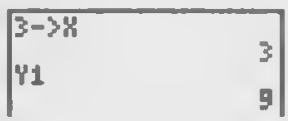
[2nd] [QUIT] [CLEAR]
[2nd] [Y-VARS] **2** [(] **3** [)] [ENTER]
[2nd] [Y-VARS] **1** [(] [2nd] [Y-VARS] **2** [(] **3**
[)] [)] [ENTER]
[2nd] [Y-VARS] **3** [(] [2nd] [{] **1, 2, 3**
[2nd] [}] [)] [ENTER]




## Evaluating Functions without Parentheses

You can also evaluate functions without using parentheses by storing a value to X.

**3** [STO▸] [X,T] [ENTER]
[2nd] [Y-VARS] **1** [ENTER]



**Only functions that are selected (turned on) are graphed. All four functions may be selected at one time.**

### Turning a Function "On" or "Off"

You can select and deselect ("turn on" and "turn off") functions on the **Y=** edit screen. The = sign on a selected function is highlighted.

To change the selection status of a function:

1. Display the **Y=** list and move the cursor to the function whose status you want to change.
2. Press ◀ to place the cursor over the = sign of the function.
3. Press ENTER to change the status. If the function was selected, it is now deselected. If it was deselected, it is now selected.

**Note:** When you enter or edit a function, it is selected automatically. When you clear a function, it is deselected.

### Selecting Functions from the Home Screen or a Program

To select functions from the Home screen or a program, begin on a blank line.

1. Press 2nd [Y-VARS], and then press ◀ to select **ON/OFF**. The **ON/OFF** menu is displayed.
2. Select the instruction you want, **FNON** or **FNOFF**. It is copied to the cursor location.
3. To turn specific functions on or off, enter the number(s) of the function(s) separated by commas.

**FNON** *function#,function#, . . .*
**FNOFF** *function#,function#, . . .*

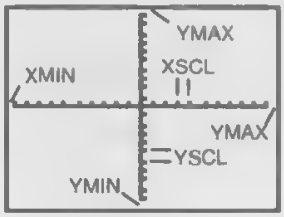
For example, in **FUNC** mode, **FNOFF 1,3** turns off functions **Y1** and **Y3**.

```
FnOFF 1,3
                DONE
```

**The Window variables determine the boundaries and other attributes of the viewing window. The window variables are shared by all graphing modes.**

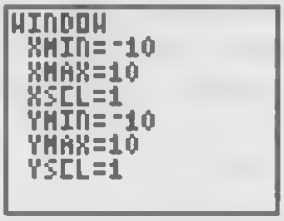
## The Viewing Window

The viewing window of the TI-80 is the portion of the coordinate plane defined by **XMIN**, **XMAX**, **YMIN**, and **YMAX**. The distance between tick marks is defined by **XSCL** for the **X** axis and **YSCL** for the **Y** axis.




## Checking the Viewing Window

Press [WINDOW] to display the current Window variable values. The values shown here are the default values.




## Changing a Window Variable Value

To change a Window variable value:

1. Press [▼] to move to the Window variable you want to change.
2. To enter a real value (which can be an expression), you may do any of the following:
   - Position the cursor, and then make the changes.
   - Press [CLEAR] to clear the value, and then enter a new value.
   - Begin entering a new value. The original value is cleared automatically when you begin typing.
3. Press [ENTER], [▼], or [▲]. If you entered an expression, it is evaluated. The new value is stored.

**XMIN** must be less than **XMAX**, and **YMIN** must be less than **YMAX**, or you will get an error message when you press [GRAPH]. To turn off the tick marks, set **XSCL=0** and **YSCL=0**.

### Leaving the Window Screen

To leave the Window screen:

- Select another screen by pressing the appropriate key, such as [GRAPH] or [Y=].
- Press [2nd] [QUIT] to return to the Home screen.

### Storing to a Window Variable from the Home Screen or a Program

To store to a Window variable from the Home screen or from a program, begin on a blank line.

1. Enter the value (which can be an expression) that you want to store.
2. Press [STO▸].
3. Press [VARS] to display the **VARS** menu.
4. Select **WINDOW...** to display the Window variables.
5. Select the Window variable. The name of the variable is copied to the cursor location.
6. Press [ENTER] to complete the instruction.

```
13->XMAX
                13
```

**Note:** You can use a Window variable in an expression by performing steps 3, 4, and 5.

### ΔX and ΔY

The variables **ΔX** and **ΔY** define the distance between the centres of two adjoining pixels on a graph (graphing accuracy).

$$\Delta X = \frac{(XMAX - XMIN)}{62} \qquad \Delta Y = \frac{(YMAX - YMIN)}{46}$$

**ΔX** and **ΔY** are not on the Window screen; however they are accessible through the **VARS WINDOW** menu. **ΔX** and **ΔY** are calculated from **XMIN**, **XMAX**, **YMIN**, and **YMAX** when a graph is displayed.

You can store values directly to **ΔX** and **ΔY** (7 and 8 on the **VARS WINDOW...** menu), in which case **XMAX** and **YMAX** are immediately calculated from **ΔX**, **XMIN**, **ΔY**, and **YMIN**.

```
.1->ΔX
                .1
.1->ΔY
                .1
```

**Pressing [GRAPH] graphs any functions selected on the Y= edit screen. The current mode settings apply, and the current values of the Window variables define the viewing window.**

### Turning the Grid Points On and Off

Grid points correspond to the axis tick marks. To turn the grid points on and off use **GRIDON** and **GRIDOFF**. The default for the TI-80 is **GRIDOFF.**

1. From the Home screen, press [2nd] [DRAW] to display the **DRAW** menu.
2. Press **9** to select **GRIDON**, or press **0** to select **GRIDOFF.**
3. Press [ENTER]. The message **DONE** is displayed.

### Displaying a New Graph

Press [GRAPH] to display the graph of the selected function or functions. (Some operations, such as **TRACE** and the Zoom instructions, display the graph automatically.) As a graph is plotted, the busy indicator is on and **X** and **Y** are updated.

### Smart Graph

When you press [GRAPH], Smart Graph displays the graph screen immediately if nothing has changed that requires the functions to be replotted since the last time the graph was displayed.

If you have changed any of the following since the graph was last displayed, pressing [GRAPH] replots the graph based on the new values:

- Changed a mode setting that affects graphs.
- Changed a function in the current picture.
- Deselected a function in the current picture.
- Changed the value of a variable in a selected function.
- Changed a Window variable or format setting.
- Cleared drawings by selecting **CLRDRAW** (Chapter 7).
- Changed or turned off a **STAT PLOT** definition (Chapter 9).

**Note: CLRDRAW** is a fast way to replot a graph.

### Overlaying Functions on a Graph

The TI-80 lets you graph one function at a time without replotting every function. For example, enter **SIN X** as **Y1** and press [GRAPH]. Then enter **COS X** as **Y2** and press [GRAPH] again. The second function is graphed on top of the original function.

# Exploring a Graph with the Free-Moving Cursor

**While a graph is displayed, you can move the free-moving cursor anywhere on the graph and display the coordinates of any location on the graph.**

## Free-Moving Cursor

You can press ◀, ▶, ▲, or ▼ to move the cursor around the graph. When you first display the graph, no cursor is visible. As soon as you press ◀, ▶, ▲, or ▼, the cursor moves from the centre of the viewing window.

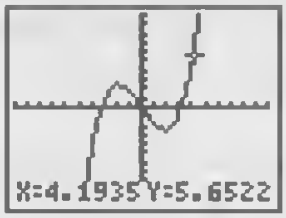
As you move the cursor around the graph, the values of the variables **X** and **Y** are updated, and the coordinate values of the cursor location are displayed at the bottom of the screen. Coordinate values generally appear in floating-decimal format. The numeric display settings on the **MODE** screen do not affect coordinate display.

To see the graph without the cursor or coordinate values, press GRAPH or CLEAR. When you press ◀, ▶, ▲, or ▼, the cursor begins to move from the same position.

## Graphing Accuracy

The free-moving cursor moves from dot to dot on the screen. When you move the cursor to a dot that appears to be "on" the function, it may be near, but not on, the function; therefore, the coordinate value displayed at the bottom of the screen is not necessarily a point on the function. To move the cursor along a function, use **TRACE** (page 4-13).

The displayed coordinate values of the free-moving cursor approximate actual math coordinates accurate to within the width/height of the dot. As **XMIN** and **XMAX** (and **YMIN** and **YMAX**) get closer together (after a **ZOOM IN**, for example), graphing accuracy increases, and the coordinate values more closely represent the math coordinates.




Free-moving cursor "on" the curve

# Exploring a Graph with TRACE

**TRACE moves the cursor from one plotted point to the next, along a function. The cursor coordinates are displayed at the bottom of the screen.**

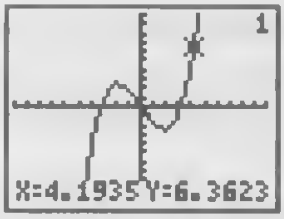
### Beginning a Trace

Press [TRACE] to begin a trace. If the graph is not displayed already, the TI-80 displays it. The cursor is on the first selected function in the **Y=** list at the middle **X** value on the screen. The number of the function appears at the top right of the display.

**Note:** If any **STAT PLOTS** are turned on, the TI-80 attempts to trace the first stat plot.

### Moving along a Function

Press [▶] and [◀] to move the cursor along the function. Each press moves the cursor from one plotted point to the next. Press [2nd] [▶] and [2nd] [◀] to move the cursor five plotted points at a time. Tracing updates and displays the values of the variables **X** and **Y**. The **Y** value is calculated from the **X** value; that is, **Y=Y*n*(X)**. If the function is undefined at an **X** value, the **Y** value is blank.




◆ Trace cursor on the curve.

If the **Y** value of a function is above or below the viewing window, the cursor disappears as you move it to that portion of the function. However, the coordinate values at the bottom of the screen indicate the cursor coordinates.

### Panning to the Left or Right

If you trace a function off the left or right edge of the screen, the viewing window automatically pans to the right or left. **XMIN** and **XMAX** are updated to correspond to the new viewing window.

**Note:** The screen does not pan if a **STAT PLOT** is on.

### QuickZoom

While tracing, you can press [ENTER] to adjust the viewing window so that the cursor location becomes the centre of a new viewing window, even if the cursor is above or below the display. This allows "panning" up and down. After QuickZoom, **TRACE** is still active.

**Moving from Function to Function**

To trace another selected function on the graph, press ▼ or ▲ to move the cursor to that function. The cursor moves to the new function at the same **X** value. The function number in the top right corner of the display changes.

The cursor movement is based on the order of the selected functions in the **Y=** list, not the appearance of the functions as graphed on the screen.

**Leaving TRACE**

To leave **TRACE**:

- Select another screen by pressing the appropriate key, such as [WINDOW] or [ZOOM].
- Press [GRAPH] or [CLEAR] to see the graph without the Trace cursor.
- Press [2nd] [QUIT] to return to the Home screen.

The Trace cursor remains in the same location if you leave **TRACE** and return, as long as Smart Graph has not caused the graph to be replotted.

**Using TRACE in a Program**

On a blank line in the program editor, press [TRACE]. The instruction **TRACE** is copied to the cursor location. When the instruction is encountered during program execution, the graph is displayed with the Trace cursor on the first selected function. As you trace, the cursor coordinate values are updated. When you finish tracing functions, press [ENTER] to resume program execution.

**Pressing [ZOOM] displays a menu that allows you to adjust the viewing window of the graph quickly in a variety of ways. All of the Zoom instructions are accessible from programs.**

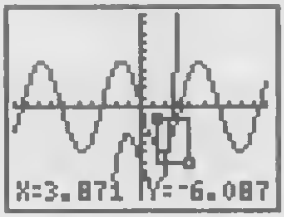
## ZOOM Menu

| ZOOM | |
|---|---|
| 1: ZBOX | Draws box to define viewing window |
| 2: ZOOM IN | Magnifies graph around cursor |
| 3: ZOOM OUT | Views more of graph around cursor |
| 4: ZDECIMAL | Sets .1 as dot size |
| 5: ZSQUARE | Sets equal sized dots on **X** and **Y** axes |
| 6: ZSTANDARD | Sets standard Window variables |
| 7: ZTRIG | Sets built-in trig Window variables |

## ZBOX

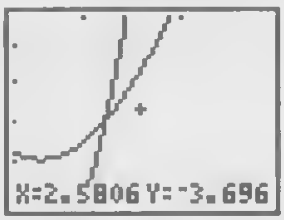
**ZBOX** lets you use the cursor to select opposite corners of a box to define a new viewing window.

1. Select **ZBOX** from the **ZOOM** menu. The different type of cursor at the centre of the screen indicates that you are using a Zoom instruction.
2. Move the cursor to any corner of the box you want to define, and then press [ENTER]. As you move the cursor away from the point just selected, you see a small square dot, indicating that the first corner is selected.
3. Move the cursor to the opposite diagonal corner of the box you want to define. As you move the cursor, the boundaries of the box change on the screen.




**Note:** Before you press [ENTER], you can cancel **ZBOX** at any time by pressing [CLEAR].

4. When the box is defined as you want it, press [ENTER] to replot the graph.




You can repeat steps 2 to 4 to do another **ZBOX**.

### ZOOM IN ZOOM OUT

**ZOOM IN** magnifies the graph around the cursor location. The **XFACT** and **YFACT** settings determine the extent of the zoom (page 4-18). The default value for both **XFACT** and **YFACT** is 4.

1. After checking or changing **XFACT** and **YFACT**, select **ZOOM IN** from the **ZOOM** menu.

   Notice the different type of cursor. It indicates that you are using a Zoom instruction.

2. Move the cursor to the point that you want as the centre of the new viewing window, and then press [ENTER].

   The TI-80 adjusts the viewing window by **XFACT** and **YFACT**, updates the Window variables, and replots the selected functions centred on the cursor location.

3. **ZOOM IN** is still turned on. To zoom in on the graph again:

   - At the same point, press [ENTER].
   - At a new point, move the cursor to the point that you want as the centre of the new viewing window, and then press [ENTER].

**ZOOM OUT** displays a greater portion of the graph, centred on the cursor location, to provide a more global view. The procedure for **ZOOM OUT** is the same as for **ZOOM IN**.

### Leaving ZOOM IN or ZOOM OUT

To leave **ZOOM IN** or **ZOOM OUT**:

- Select another screen by pressing the appropriate key, such as [TRACE] or [GRAPH].
- Press [2nd] [QUIT] to return to the Home screen.

**ZDECIMAL** — **ZDECIMAL** replots the functions immediately, updates the window variables to preset values that set **ΔX** and **ΔY** equal to **.1**, and defines the **X** and **Y** value of each pixel as one decimal.

| | |
|---|---|
| **XMIN = -3.1** | **YMIN = -2.3** |
| **XMAX = 3.1** | **YMAX = 2.3** |
| **XSCL = 1** | **YSCL = 1** |

**ZSQUARE** — **ZSQUARE** replots the functions immediately, redefining the window based on the current Window variables, but adjusted in only one direction so that **ΔX=ΔY**. This makes the graph of a circle look like a circle. **XSCL** and **YSCL** remain unchanged. The midpoint of the current graph (not the intersection of the axes) becomes the midpoint of the new graph.

**ZSTANDARD** — **ZSTANDARD** replots the functions immediately, updating the Window variables to the standard values:

| | |
|---|---|
| **XMIN = -10** | **YMIN = -10** |
| **XMAX = 10** | **YMAX = 10** |
| **XSCL = 1** | **YSCL = 1** |

**ZTRIG** — **ZTRIG** replots the functions immediately, updating the Window variables to preset values appropriate for plotting trig functions. In **RADIAN** mode. These are:

| | |
|---|---|
| **XMIN = -(31/12)π (-8.115781..)** | **YMIN = -2 (-2)** |
| **XMAX = (31/12)π (8.1157810..)** | **YMAX = 2 (2)** |
| **XSCL = (π/2) (1.5707963..)** | **YSCL = 1 (1)** |

# Setting the ZOOM Factors

**The Zoom factors, XFACT and YFACT, determine the extent of the change for the viewing window created by ZOOM IN or ZOOM OUT on a graph.**

### Zoom Factors

Zoom factors are positive numbers (not necessarily integers) greater than or equal to 1. They define the magnification or reduction factor used to **ZOOM IN** or **ZOOM OUT** around a point.

### Checking Zoom Factors

To review the current values of the Zoom factors (**XFACT** and **YFACT**):

1. Press [VARS], and then press **1** to select **WINDOW...**.
2. Press **9** to select **XFACT** or **0** to select **YFACT**. **XFACT** or **YFACT** is copied to the cursor location.
3. Press [ENTER]. The Zoom factor is displayed.


### Setting Zoom Factors from the Home Screen or a Program

To set the zoom factors **XFACT** and **YFACT** from the Home screen or a program:

1. Enter the factor, and then press [STO▸].
2. Press [VARS], and then press **1** to select **WINDOW...**.
3. Press **9** to select **XFACT** or **0** to select **YFACT**. **XFACT** or **YFACT** is copied to the cursor location on the Home screen.
4. Press [ENTER] to store the Zoom factor to the variable.

# Chapter 5: Parametric Graphing

**This chapter describes how to graph parametric equations on the TI-80. Before going on to parametric graphing, you should be familiar with Chapter 4, Function Graphing.**

**Chapter Contents**

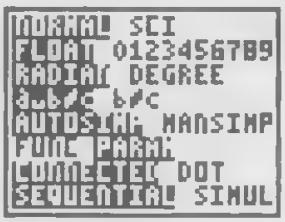
# Getting Started: Path of a Ball

**Getting Started is a fast-paced introduction. Read the chapter for details.**

**Graph the parametric equation that describes the position of a ball kicked at an angle of 60° with an initial velocity of 15 meters per second. (Ignore air resistance.) What is the maximum height? When does the ball strike the ground?**

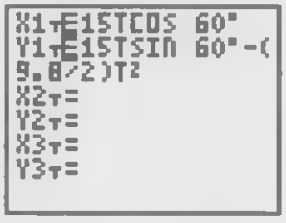
1. Press [MODE]. Press [▼] [▼] [▼] [▼] [▼] [▶] [ENTER] to select **PARAM** mode.

   For initial velocity $v_0$ and angle $\theta$, the horizontal component of the ball as a function of time is $X(t) = t\, v_0 \cos \theta$. The vertical component is $Y(t) = t\, v_0 \sin \theta - (g/2)\, t^2$. The gravity constant g is 9.8 m/sec$^2$.



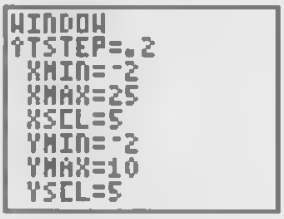

2. Press [Y=]. Press **15** [X,T] [COS] **60** [2nd] [ANGLE] **1** (to select °) [ENTER] to define the **X** portion of the parametric equation in terms of **T**.

3. Press **15** [X,T] [SIN] **60** [2nd] [ANGLE] **1** (to select °) [−] [(] **9.8** [÷] **2** [)] [X,T] [$x^2$] [ENTER] to define the **Y** portion.




4. Press [WINDOW]. Enter the Window variables appropriate for this problem.

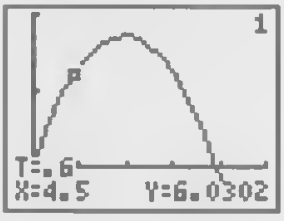
| | | |
|---|---|---|
| **TMIN=0** | **XMIN=-2** | **YMIN=-2** |
| **TMAX=3** | **XMAX=25** | **YMAX=10** |
| **TSTEP=.2** | **XSCL=5** | **YSCL=5** |




5. Press [TRACE] to graph the position of the ball as a function of time.

   Tracing begins at **TMIN**. As you press [▶] to trace the curve, the cursor follows the path of the ball over time. The values for **X** (distance), **Y** (height), and **T** (time) are displayed at the bottom of the screen.

   The maximum height is approximately 8.6 meters. The ball strikes the ground in approximately 2.6 seconds.



# Defining and Displaying a Parametric Graph

**Parametric equations consist of an X component and a Y component, each expressed in terms of the same independent variable T. They are often used to graph equations over time. Up to three pairs of parametric equations can be defined and graphed at a time.**

### Defining a Parametric Graph

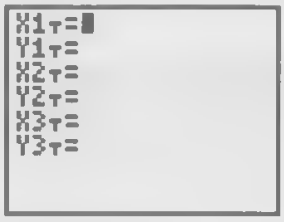
The steps for defining a parametric graph are the same as those for defining a function graph (page 4-3). The differences are listed below.

### Setting Parametric Graph Modes

Press [MODE] to display the mode settings. To graph parametric equations, you must select **PARAM** before you enter Window variables or enter the components of parametric equations. Normally, you should also select **CONNECTED** to obtain a more meaningful parametric graph.

### Displaying Parametric Equations

After selecting **PARAM** mode, press [Y=] to display the parametric **Y=** edit screen.




On this screen, you display and enter both the **X** and **Y** components. The TI-80 allows three parametric equations, each defined in terms of **T**.

### Defining Parametric Equations

Follow the same procedures as for function graphing (pages 4-5 to 4-6) to enter the two components that define a new parametric equation.

- You must define both the **X** and **Y** components in a pair.
- The independent variable in each component is **T**. You may press [X,T], rather than pressing [ALPHA] [T], to enter the parametric variable **T**. (**PARAM** mode defines the independent variable as **T**.)

### Selecting Parametric Equations

The TI-80 graphs only those parametric equations you select. The highlighted = sign on both components of an equation indicates that the equation is selected.

You may select any or all of the equations on the parametric **Y=** edit screen.

To change the selection status of a parametric equation, press [◄] to move the cursor onto the = sign of either the **X** or **Y** component and press [ENTER]. The status of both the **X** and **Y** components changes.

**Note:** When you enter both components of an equation or edit either component, that equation is selected automatically.

### Setting Window Variables

Press [WINDOW] to display the current Window variable values. The Window variables define the viewing window. The values shown are the standard values in **RADIAN** mode.

| | |
|---|---|
| TMIN=0 | Smallest T value to evaluate. |
| TMAX=6.283185307 | Largest T value to evaluate. |
| TSTEP=.1308996938996 | T value increment. |
| XMIN=-10 | Smallest X value to be displayed. |
| XMAX=10 | Largest X value to be displayed. |
| XSCL=1 | Spacing between X tick marks. |
| YMIN=-10 | Smallest Y value to be displayed. |
| YMAX=10 | Largest Y value to be displayed. |
| YSCL=1 | Spacing between Y tick marks. |

To change a Window variable value, follow the steps given for function graphing (page 4-8).

**Note:** You may want to change the **T** variable values to ensure that sufficient points are plotted.

### Displaying a Graph

When you press GRAPH, the TI-80 plots the selected parametric equations. It evaluates both the **X** and the **Y** component for each value of **T** (from **TMIN** to **TMAX** in intervals of **TSTEP**) and then plots each point defined by **X** and **Y**. The Window variables define the viewing window.

As a graph is plotted, the TI-80 updates **X**, **Y**, and **T**.

**Note:** Smart Graph applies to parametric graphs also page 4-11.

### VARS WINDOW and Y-VARS Menus

By means of the **VARS WINDOW** and **Y-VARS** menus, you can:

- Access functions by using the name of the component of the equation as a variable.
- Select or deselect parametric equations from a program, using the **FNON** and **FNOFF** commands (page 4-8).
- Store parametric equations.
- Store values directly to Window variables.

```
π/24->TSTEP
      .1308996939
"10cos 2(πT/6)"
->X1T
                DONE
```

# Exploring a Parametric Graph

**As in Function graphing, three tools are available for exploring a graph: the free-moving cursor, tracing, and zooming.**

### Free-Moving Cursor

The free-moving cursor works the same in parametric graphing as in function graphing (page 4-12).

### Tracing a Parametric Graph

Pressing [TRACE] puts the Trace cursor on the first selected equation, at **TMIN**. You can then trace along the equation.

| | |
|---|---|
| [◀] or [▶] | Moves the cursor one **TSTEP** at a time. |
| [2nd] [◀] or [▶] | Moves the cursor five **TSTEP**s at a time. |
| [▲] or [▼] | Changes to the previous or next equation. The equation number is displayed in the top right of the display. |
| [CLEAR] | Cancels tracing. |

For each value of **T**, the calculator displays values for **X** and **Y**.

The values for **X**, **Y**, and **T** are updated as you move the Trace cursor. If the cursor moves off the top or bottom of the screen, the coordinate values continue to change appropriately.

The Trace cursor remains in the same location if you leave **TRACE** and return, unless Smart Graph replots the graph.

QuickZoom is available in parametric graphing, but panning is not (page 4-13).

### Zooming in on a Parametric Graph

Pressing [ZOOM] works the same in parametric graphing as in function graphing (page 4-15).

Parametric graphing adds the Window variables **TMIN**, **TMAX**, and **TSTEP**. These variables are not affected by zooming unless you select **ZSTANDARD**, where **TMIN** = 0, **TMAX** = 6.283185307 ($2\pi$), and **TSTEP** = .1308996938996 ($\pi/24$).

# Chapter 6: Tables

**This chapter describes how to use tables on the TI-80. A table evaluates the selected functions from the Y= list and displays each value for the independent variable along with the evaluated value for each corresponding dependent variable.**

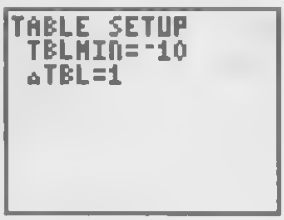
**Chapter Contents**

# Getting Started: Roots of a Function

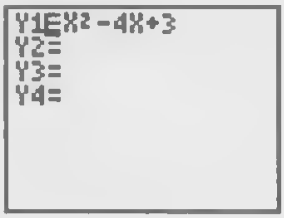
**Getting Started is a fast-paced introduction. Read the chapter for details.**

**Evaluate the function $Y=X^2-4X+3$ at each integer between -10 and 10. How many sign changes are there and where do they occur?**

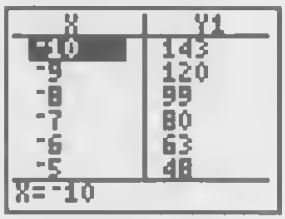
1. If necessary, select **FUNC** from the **MODE** menu. Press [2nd] [TblSet] to display the **TABLE SETUP** screen. Press [(-)] **10** to set **TBLMIN=-10**. Leave **ΔTBL=1**.



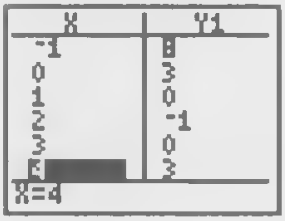

2. Press [Y=] [X,T] [$x^2$] [−] **4** [X,T] [+] **3** to enter the function **Y1=X²–4X+3**.




3. Press [2nd] [TABLE] to display the table screen.




4. Press [▼] repeatedly to view the changes in the value of **Y1**.



# Defining the Independent Variable

**The independent variable for a table is the independent variable in the current graphing mode (X for FUNC mode and T for PARAM mode). You define the minimum value and the incremental value for the independent variable on the TABLE SETUP screen.**

### TABLE SETUP Screen

To display the **TABLE SETUP** screen, press [2nd] [TblSet]. The default values are shown below.

```
TABLE SETUP
 TBLMIN=0
 ΔTBL=1
```

### TBLMIN and ΔTBL

**TBLMIN** (table minimum) defines the initial value for the independent variable: **X** (**FUNC** mode) or **T** (**PARAM** mode).

**ΔTBL** (table step) defines the increment for the independent variable.

To change **TBLMIN** and **ΔTBL**, simply enter the values at the flashing cursor. To move between **TBLMIN** and **ΔTBL**, press [▼] and [▲].

### Setting Up a Table from the Home Screen or a Program

You can also store values to **TBLMIN** and **ΔTBL** from the Home screen or a program. The variable names are on the **VARS TABLE...** menu.

To change **TBLMIN** or **ΔTBL** from the Home screen or a program, begin on a blank line.

1. Enter the value for **TBLMIN** or **ΔTBL**.
2. Press [STO▸].
3. Press [VARS] to display the **VARS** menu.
4. Press **3** to select **TABLE...**.
5. Select the table variable (**TBLMIN** or **ΔTBL**). The name of the variable is copied to the cursor location.
6. Press [ENTER] to store the value for the table variable.

```
-10->TBLMIN
                -10
1->ΔTBL
                  1
```

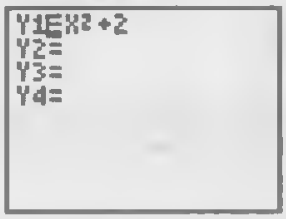
## Defining the Dependent Variable

**The selected functions from the Y= list define the dependent variables. You can have as many dependent variables as there are functions in the current graphing mode (four for FUNC mode and six for PARAM mode).**

### From the Y= Editor

Enter the functions to define the dependent variables in the **Y=** editor.

**FUNC Mode**




**PARAM Mode**


In **PARAM** mode, you must define both components of the parametric equation (Chapter 5).

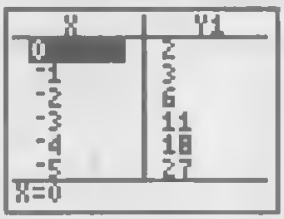
Only functions that are selected are displayed in the table. (When = is highlighted, the function is selected.) You can select and deselect functions from the **Y=** list, from the Home screen, or from a program. (Refer to page 4-8 for information on selecting and deselecting.)

**The table displays up to six values for the independent variable, along with the six corresponding values of each dependent variable, one at a time. Once the table is displayed, you can press ◄, ▲, ►, and ▼ to move around and scroll through the table, displaying other independent and dependent values.**

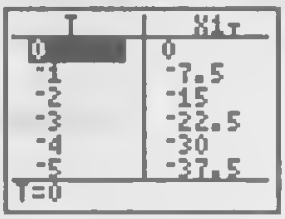
## The Table

Press 2nd [TABLE] to display the table screen.

**FUNC Mode**




**PARAM Mode**

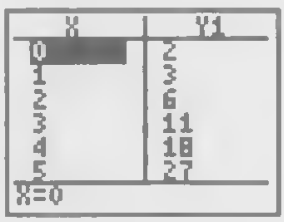


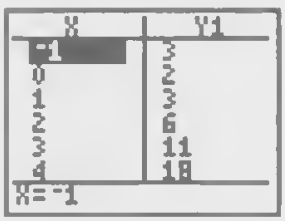

The top line displays the name of the independent variable (**X** for **FUNC** mode; **T** for **PARAM** mode) and one dependent variable (**Y*n*** for **FUNC** mode; **X*n*T** or **Y*n*T** for **PARAM** mode). The bottom line displays the full value of the current cell, which is indicated by the rectangular cursor. The centre portion is used to display the values of the variables, abbreviated to six digits if necessary.

## Displaying More Independent Values

Press ▲ and ▼ to display additional values for the independent variable and the values for one corresponding dependent variable.

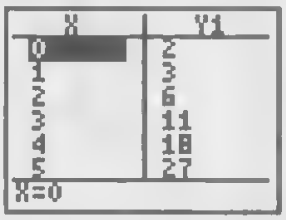
**Note:** You can scroll "back" from the value entered for **TBLMIN**. As you scroll, **TBLMIN** is updated automatically to the value shown on the top line of the table. In the example below, **TBLMIN=0**, **ΔTBL=1** and **$Y_1=X^2+2$** generates and displays values of **X**=0, . . ., 5. You can press ▲ to scroll backward and display the table for **X**=1, . . .,4.

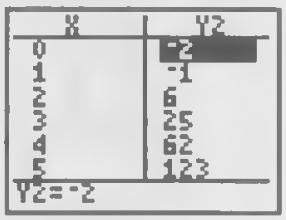


**Displaying Other Dependent Variables**

If you have defined and selected more than one function, press ▶ to display other dependent variables. In the example below, **TBLMIN=0**, **ΔTBL=1**, **$Y_1=X^2+2$**, and **$Y2=X^3-2$**. You can press ▶ ▶ to see the values for **Y2**.



# Chapter 7: Draw Operations

**This chapter describes how to use the DRAW operations of the TI-80. Before using the DRAW operations, you should be familiar with Chapter 4, Function Graphing.**

**Chapter Contents**

# Getting Started: Shading a Graph

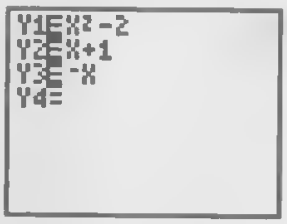
**Getting Started is a fast-paced introduction. Read the chapter for details.**

**Shade the area below the function $Y=X^2-2$ and above the functions $Y=X+1$ and $Y=-X$.**

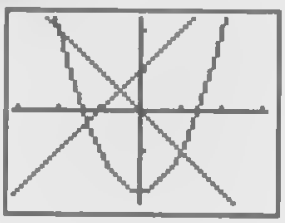
1. If necessary, select **FUNC** mode. Press [Y=] and enter the functions:

   **Y1**=[X,T] [$x^2$] [−] **2** [ENTER]
   **Y2**=[X,T] [+] **1** [ENTER]
   **Y3**=[(-)] [X,T] [ENTER]

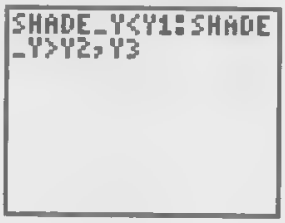
   (Be sure that **Y4** is cleared or turned off.)



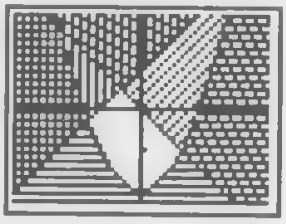

2. Press [ZOOM] **4** to select the **ZDECIMAL** viewing window, clear any existing drawings, and display the viewing window and graph.



3. Press [2nd] [QUIT] to return to the Home screen.
4. Press [2nd] [DRAW] **7** to select **SHADE_Y<**, which is copied to the Home screen.
5. Press [2nd] [Y-VARS] **1** (to select **Y1**).
6. Press [2nd] [:] to add another instruction to this line.
7. Press [2nd] [DRAW] **6** to select **SHADE_Y>**, which is copied to the Home screen.
8. Press [2nd] [Y-VARS] **2** (to select **Y2**) [,] [2nd] [Y-VARS] **3** (to select **Y3**).




9. Press [ENTER] to see the functions and shading on the graph.



**To display the DRAW DRAW menu, press [2nd] [DRAW]. What happens when you select an item from this menu depends on whether or not a graph is displayed when you access the menu, as described under each operation.**

## DRAW DRAW Menu

| DRAW POINTS | |
|---|---|
| 1:CLRDRAW | Clears all drawn elements. |
| 2:LINE( | Draws a line between two points. |
| 3:HORIZONTAL | Draws a horizontal line. |
| 4:VERTICAL | Draws a vertical line. |
| 5:DRAWF | Draws a function. |
| 6:SHADE_Y> | Shades an area. |
| 7:SHADE_Y< | Shades an area. |
| 8:SHADE( | Shades an area. |
| 9:GRIDON | Turns the graph grid on. |
| 0:GRIDOFF | Turns the graph grid off. |

See page 7-11 for an explanation of **CLRDRAW**.

## Before Drawing on a Graph

Because Draw operations draw over the graph of currently selected functions, you may want to do one or more of the following before drawing on a graph:

- Change the mode settings.
- Enter or edit functions in the **Y=** list.
- Select or deselect functions in the **Y=** list.
- Change Window variable values.
- Turn **STAT PLOTS** on or off.
- Clear existing drawings with **CLRDRAW** (page 7-11).

## Drawing on a Graph

Draw operations can draw on **FUNC** and **PARAM** graphs. The coordinates for all Draw instructions are always the X-coordinate and Y-coordinate values of the display.

You can use most of the operations from the **DRAW DRAW** and **DRAW POINTS** menus to draw directly onto a graph, using the cursor to identify coordinates; or you can execute these instructions from the Home screen or a program.

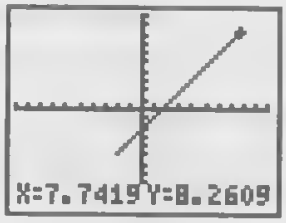
## Drawing Lines

**While a graph is displayed, LINE( lets you use the cursor to define a line on the graph. If a graph is not displayed, the instruction is copied to the Home screen.**

### Directly on a Graph

To define a line directly on a graph:

1. When a graph is displayed, select **LINE(** from the **DRAW DRAW** menu (item 2).
2. Position the cursor at the beginning point of the line you want to draw. Press [ENTER].
3. Move the cursor to the end point of the line you want to draw. The line is displayed as you move the cursor. Press [ENTER].



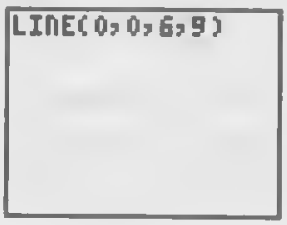

To continue to draw lines, repeat steps 2 and 3. To cancel **LINE(**, press [CLEAR].

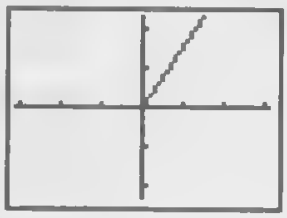
### From the Home Screen or a Program

**LINE(** (**DRAW DRAW**, item 2) draws a line between the coordinates (*X1*,*Y1*) and (*X2*,*Y2*). The values may be entered as expressions.

**LINE(***X1*,*Y1*,*X2*,*Y2*)

For example, enter **LINE(0,0,6,9)** on the Home screen and then press [ENTER].



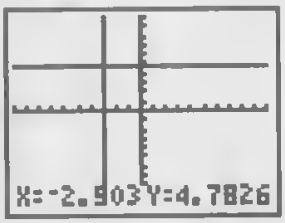
# Drawing Horizontal and Vertical Lines

**While a graph is displayed, HORIZONTAL and VERTICAL let you define lines on the graph using the cursor. If a graph is not displayed, the instruction is copied to the Home screen.**

## Directly on a Graph

To draw horizontal and vertical lines directly on a graph:

1. When a graph is displayed, select **HORIZONTAL** (item 3) or **VERTICAL** (item 4) from the **DRAW DRAW** menu.
2. A line is displayed that moves as you move the cursor. Position the cursor where you want to draw the line. Press [ENTER]. The line is drawn on the graph.




To continue to draw lines, repeat steps 1 and 2. To cancel **HORIZONTAL** or **VERTICAL**, press [CLEAR].

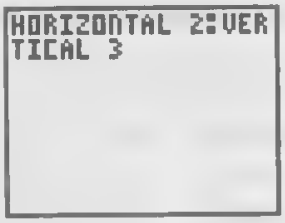
## From the Home Screen or a Program

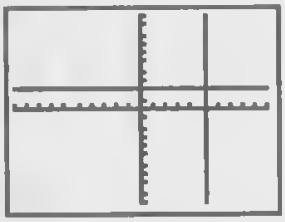
**HORIZONTAL** (**DRAW DRAW**, item 3) draws a horizontal line at **Y**=*Y* (which can be an expression, but not a list).

**HORIZONTAL** *Y*

**VERTICAL** (**DRAW DRAW**, item 4) draws a vertical line at **X**=*X* (which can be an expression, but not a list).

**VERTICAL** *X*



**Note:** In the example above, the horizontal line is drawn first, and then the vertical line.

# Drawing a Function

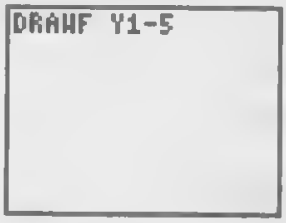
**DRAWF (draw function) draws a function on the current graph. DRAWF must be entered on the Home screen or in the program editor.**

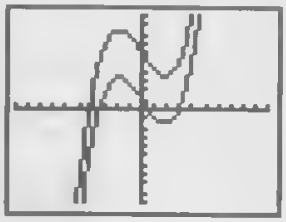
## Drawing a Function

**DRAWF** (draw function, **DRAW DRAW**, item 5) is not an interactive operation. It draws the specified expression as a function in terms of **X** on the current graph.

**DRAWF** *expression*

For example, if **Y1=.2X$^3$–2X+6** is the only selected function, **DRAWF Y1–5** plots **Y1** and then draws the function **Y1–5** when you press [ENTER].



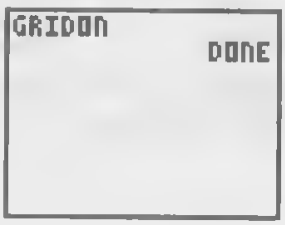



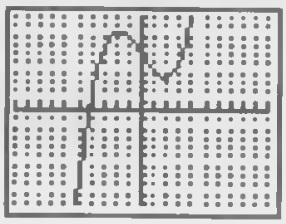
**Note:** You cannot trace on **DRAWF** functions.

## Using GRIDON and GRIDOFF

**GRIDON** (**DRAW DRAW**, item 9) and **GRIDOFF** (**DRAW DRAW**, item 0) set graphs to be drawn with the grid points on or off, respectively. The grid points correspond to the tick marks on the axes. The default for the TI-80 is **GRIDOFF**.

1. From the Home Screen, press [2nd] [DRAW] to display the **DRAW** menu.
2. Press **9** to select **GRIDON**, or press **0** to select **GRIDOFF**.
3. Press [ENTER]. The message **DONE** is displayed. The next time the graph is displayed, the grid points will be on if you selected **GRIDON**, or off if you selected **GRIDOFF**.



# Shading Areas on a Graph

**There are three shading instructions on the DRAW DRAW menu: SHADE_Y>, SHADE_Y<, and SHADE(. These instructions are not interactive; they must be entered on the Home screen or in the program editor.**

## Shading Areas above a Function

**SHADE_Y>** (**DRAW DRAW**, item 6) takes up to four arguments (functions of **X**).

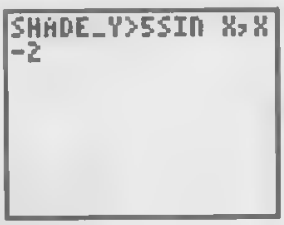
**SHADE_Y>***func*
**SHADE_Y>***func1*,...,*func4*

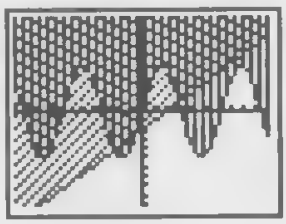
When executed, **SHADE_Y>** plots the specified function(s) on the graph and shades the area above the function with a pattern.

The patterns are automatically assigned in the following order.

| | |
|---|---|
| Function 1 | Vertical pattern. |
| Function 2 | Diagonal pattern, bottom left to top right. |
| Function 3 | Diagonal pattern, top left to bottom right. |
| Function 4 | Horizontal pattern. |

When you specify multiple functions, the shading is done sequentially.



### Shading Areas below a Function

**SHADE_Y<** (**DRAW DRAW**, item 7) takes up to four arguments (functions of **X**).

**SHADE_Y<***func*
**SHADE_Y<***func1,...,func4*

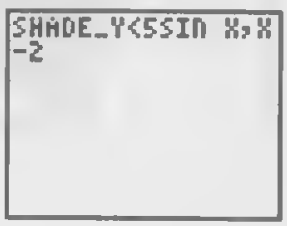
When executed, **SHADE_Y<** plots the specified function(s) on the graph and shades the area below the function with a pattern.

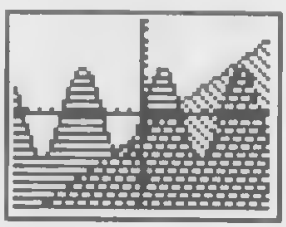
The patterns are automatically assigned in the following order.

| | |
|---|---|
| Function 1 | Horizontal pattern. |
| Function 2 | Diagonal pattern, top left to bottom right. |
| Function 3 | Diagonal pattern, bottom left to top right. |
| Function 4 | Vertical pattern. |

Note that the patterns are assigned in reverse order from the **SHADE_Y>** patterns.

When you specify multiple functions, the shading is done sequentially.



## Shading a Graph

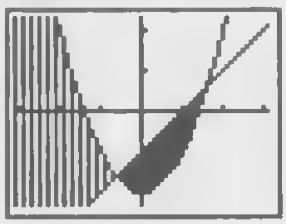
**SHADE(** (**DRAW DRAW**, item 8) shades the area on a graph that is below one specified function and above another, between two X values. **SHADE(** is not an interactive operation. It draws *lowerfunc* and *upperfunc* in terms of X on the current graph and shades the area that is specifically above *lowerfunc* and below *upperfunc*. Only the areas where *lowerfunc* < *upperfunc* are shaded.

You can specify the shading resolution (an integer between 1 and 9). If none is specified, 1 is used. *resolution*=1 shades every pixel. *resolution*=2 shades every second pixel. *resolution*=3 shades every third pixel, and so on.

Optionally, you can specify *Xleft* (the left boundary) and *Xright* (the right boundary) for the shaded area. If *Xleft* or *Xright* is not specified, **XMIN** and **XMAX** are used.

**SHADE(***lowerfunc*,*upperfunc*)
**SHADE(***lowerfunc*,*upperfunc*,*resolution*)
**SHADE(***lowerfunc*,*upperfunc*,*resolution*,*Xleft*)
**SHADE(***lowerfunc*,*upperfunc*,*resolution*,*Xleft*,*Xright*)

```
SHADE(X²-2,X-1)
SHADE(X-1,X²-2,2
,XMIN,0)
```



# Drawing Points

**To display the DRAW POINTS menu, press [2nd] [DRAW] [▶]. What happens when you select an item from this menu depends on whether or not a graph is displayed when you access the menu, as described under each operation.**

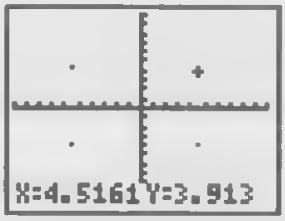
## DRAW POINTS Menu

| DRAW POINTS | |
|---|---|
| 1: PT-ON( | Turn on a point. |
| 2: PT-OFF( | Turn off a point. |
| 3: PT-CHANGE( | Toggle a point on or off. |

## Drawing a Point Directly on a Graph

To draw points directly on a graph:

1. When a graph is displayed, select **PT-ON(** from the **DRAW POINTS** menu (item 1).
2. Position the cursor at the location on the display where you want to draw the point. Press [ENTER]. The point is drawn.




To continue to draw points, repeat step 2. To cancel **PT-ON(**, press [CLEAR].

**PT-OFF**
**PT-CHANGE(**

The procedures for using **PT-OFF(** (point off, **DRAW POINTS**, item 2) to turn off (erase) a point and **PT-CHANGE(** (point change) to toggle (reverse) a point on and off are the same as for **PT-ON(**.

**From the Home Screen or a Program**

When you use these instructions from the Home screen or a program, you must specify the X- coordinate and the Y- coordinate of the point as arguments for the instructions.

**PT-ON(** turns on the point at (**X**=*X*,**Y**=*Y*). **PT-OFF(** turns the point off, and **PT-CHANGE(** toggles the point between on and off.

**PT-ON(***X*,*Y*)
**PT-OFF(***X*,*Y*)
**PT-CHANGE(***X*,*Y*)

```
PT-ON(2,5)
PT-OFF(3,0)
PT-CHANGE(2,5)
```

# Clearing a Drawing

**All points, lines, and shading drawn on a graph with DRAW operations are temporary. They remain only until you execute a CLRDRAW (clear drawing) instruction or a change prompts Smart Graph to replot the graph, at which time all drawn elements are erased.**

### When a Graph Is Displayed

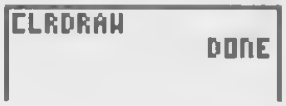
To clear drawings from the currently displayed graph, select **CLRDRAW** from the **DRAW DRAW** menu (item 1). The current graph is plotted and displayed immediately with no drawn elements.

Note that **CLRDRAW** is a quick way to replot the current graph, in addition to clearing the current drawings.

### From the Home Screen or a Program

Begin on a blank line on the Home screen or in the program editor. Select **CLRDRAW** from the **DRAW DRAW** menu (item 1). The instruction is copied to the cursor location.

When the instruction is executed, it clears all drawings from the current graph and displays the message **DONE**. The next time you display the graph, all drawn points, lines, and shaded areas will be gone.



# Chapter 8: Lists

**This chapter describes the list features of the TI-80. The TI-80 can store up to six lists. A list, depending on available memory, may contain up to 99 elements.**

**Chapter Contents**

# Getting Started: Generating a Sequence

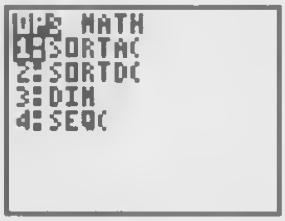
**Getting Started is a fast-paced introduction. Read the chapter for details.**

**Calculate the first eight terms of the sequence $1/A^2$ and display them in fraction form.**

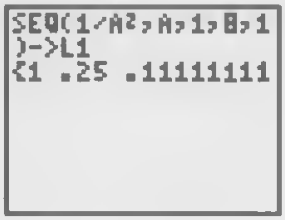
The **SEQ(** function returns a list of values based on five arguments: an expression, a variable to be incremented, a beginning value, an ending value, and an increment.

For this example, the beginning value is 1, the ending value is 8, and the increment is 1.

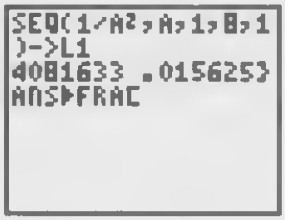
1. Begin on a blank line on the Home screen. Press [2nd] [LIST] to display the **LIST OPS** menu.



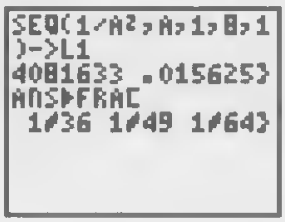

2. Press **4** to select **SEQ(**. The function name is copied to the cursor location on the Home screen.
3. Press **1** [÷] [ALPHA] **A** [$x^2$] [,] [ALPHA] **A** [,] **1** [,] **8** [,] **1** [)] [STO▸] [2nd] [L1]. Press [ENTER] to generate the list and store it in **L1**. The list is displayed on the Home screen.




4. Use [▸] to scroll through the list to see all of the elements.
5. Press [FRAC] **4** (to select ▸**FRAC**). On the Home screen, **ANS** is typed automatically, followed by ▸**FRAC**.




6. Press [ENTER] to show the sequence in fraction form. Use [▸] to scroll through the list to see all of the elements.



# About Lists

**The TI-80 has six list variables: L1, L2, L3, L4, L5, and L6. On the Home screen or in a program, you can use, enter, store, and display lists. The list names are on the keyboard. A list may contain a maximum of 99 elements.**

## Using a List in an Expression

To use a list in an expression, you may:

- Use the name of the list (**L1**, **L2**, **L3**, **L4**, **L5**, or **L6**) in the expression.

```
5+L1
```

- Enter the list directly in the expression.

```
5+{1,2,3}
```

## Entering a List In an Expression

1. Press [2nd] [{] to indicate the beginning of the list.
2. Enter a value (which can be an expression) for each element in the list, separated by commas.
3. Press [2nd] [}] to indicate the end of the list.

```
2×{1,2+3,4²}
        {2 10 32}
```

The expression is evaluated when the entry is executed. Commas are required on entry to separate elements, but they are not displayed on output.

## Saving a List In Memory

You can save a list in memory in two ways:

- Enter the list in the **STAT** list editor (Chapter 9).
- Enter the list on a blank line on the Home screen or in a program, press [STO▸], and then enter the name of the list (**L1**, **L2**, **L3**, **L4**, **L5**, or **L6**).

```
2×{1,2+3,4²}->L6
        {2 10 32}
```

## Copying One List to Another

To copy a list, store it to another list.

```
L6->L5
        {2 10 32}
```

**Displaying a List on the Home Screen**

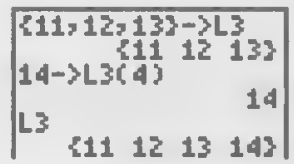
To display the contents of a list on the Home screen, enter the name of the list and press [ENTER].

An open brace ({) ithout a corresponding close brace (}) indicates that a list is too long to be displayed in its entirety. Press [▶] and [◀] to display the rest of the list.


**Storing to or Recalling a List Element**

You can store a value to or recall a value from a specific list element. Enter the name of the list, followed by the number of the element in parentheses. You can store to any element within the currently defined list dimensions, or one beyond.

*listname*(*element*)




You can also edit a list by means of the **STAT** list editor (Chapter 9).

**Lists In Graphing**

You can use a list in a **Y=** expression. However, the list must be used in such a way that it resolves to a single value; for example, **Y1=X*SUM(1/(1.1^{1,2,3}))**.

**Note:** Unlike the TI-82 and TI-85, you cannot use a list to graph a family of curves.

**Notes about Using Math Functions with Lists**

A list can be used to input several values for certain functions. (Other chapters and Appendix A state when a list is valid.) The function is evaluated for each element in the list, and a list is returned.

- If a list is used with a function, the function must be valid for every element in the list.

```
1/{1,0,-1}
```

◆ This returns an error because 1 is divided by 0.

- If two lists are used with a two-argument function, the lengths of the lists must be the same. The answer is a list in which each element is calculated by evaluating the function according to the corresponding elements in the lists.

```
{1,2,3}+{4,5,6}
                {5 7 9}
```

- If a list and a value are used with a two-argument function, the value is used with each element in the list.

```
{1,2,3}+4
              {5 6 7}
4{1,2,3}
            {4 8 12}
```

# LIST OPS Operations

**Press [2nd] [LIST] to display the list operations on the LIST OPS menu.**

**LIST OPS Menu**

| OPS MATH | |
|---|---|
| 1: SORTA( | Sorts lists in ascending order. |
| 2: SORTD( | Sorts lists in descending order. |
| 3: DIM | Accesses the list dimension. |
| 4: SEQ( | Creates a sequence. |

**Note: SORTA(** and **SORTD(** are the same as **SORTA(** and **SORTD(** on the **STAT EDIT** menu.

**SORTA(**
**SORTD(**

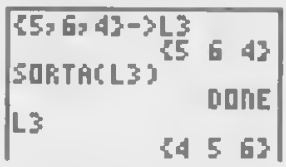
**SORTA(** (sort ascending, **LIST OPS**, item 1) and **SORTD(** (sort descending, **LIST OPS**, item 2) have two uses.

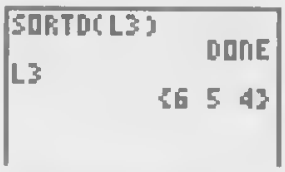
- With one list name, they sort the elements of an existing list and update the list in memory.
- With two to six list names, they sort the first list and then sort the remaining lists as dependent lists, placing their elements in the same order as their corresponding elements in the first list. This allows you to keep sets of related data in the same order when you sort lists.

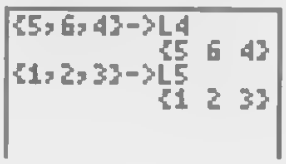
All of the lists to be sorted must be the same length. The sorted lists are updated in memory.

**Note:** You can reference a specific list only once in these instructions.

**SORTA(***listname*)
**SORTA(***keylistname*,*dependlist1*,*dependlist2*, . . .)
**SORTD(***listname*)
**SORTD(***keylistname*,*dependlist1*,*dependlist2*, . . .)




### Accessing List Dimensions with DIM

**DIM** (dimension, **LIST OPS**, item 3) returns the length (number of elements) of the specified list.

**DIM** *list*

```
DIM {1,3,5,7}
                4
```

### Creating a List with DIM

**DIM** is used with STO▸ to create a new list with a specified number of elements. The elements of the new list are zeros.

*length*→**DIM** *listname*

```
3->DIM L2
                3
L2
          {0 0 0}
```

### Redimensioning a List with DIM

**DIM** is also used with STO▸ to redimension an existing list.

- The elements in the old list that are within the new dimension are not changed.
- Any elements in the old list that are outside the new dimension are eliminated.
- Any additional elements that are created are zeros.

*length*→**DIM** *listname*

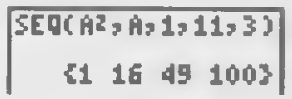
```
{1,3,5,7}->L4
          {1 3 5 7}
5->DIM L4:L4
        {1 3 5 7 0}
3->DIM L4:L4
            {1 3 5}
```

**SEQ(**

**SEQ(** (sequence, **LIST OPS**, item 4) requires five arguments: an expression, a variable to be incremented, a beginning value, an ending value, and an increment. **SEQ(** returns a list in which each element is the value of *expression* evaluated at *increments* for *variable* from *begin* to *end*.

**SEQ(***expression*,*variable*,*begin*,*end*,*increment*)

The variable need not be defined in memory. The increment can be negative.




**SEQ(** can be used to generate a list of index numbers. This kind of list can be useful in data analysis.

**Pressing 2nd [LIST] ▶ accesses the list math operations on the LIST MATH menu.**

**LIST MATH Menu**

```
OPS  MATH
1:MIN(        Returns minimum element of a list.
2:MAX(        Returns maximum element of a list.
3:MEAN(       Returns mean of a list.
4:MEDIAN(     Returns median of a list.
5:SUM         Returns sum of all elements in a list.
6:PROD        Returns product of all elements in a list.
```

Note: **MIN(** and **MAX(** are the same as **MIN(** and **MAX(** on the **MATH NUM** menu.

**MIN(**
**MAX(**

**MIN(** (minimum, **LIST MATH**, item 1) or **MAX(** (maximum, **LIST MATH**, item 2) returns the smallest or largest element of the specified list. If two lists are compared, it returns a list of the smaller or larger of each pair of elements in the two lists.

**MIN(***list*) **MAX(***list*)
**MIN(***listA*,*listB*) **MAX(***listA*,*listB*)

```
MIN({1,2,3})
                1
MAX({1,2,3},{3,2
,1})
          {3 2 3}
```

**MEAN(**
**MEDIAN(**

**MEAN(** (**LIST MATH**, item 3) returns the mean value of the list. **MEDIAN(** (**LIST MATH**, item 4) returns the median value of the list.

**MEAN(***list*) or **MEDIAN(***list*)

If a second list is given, it is interpreted as the frequency of the elements in the list.

**MEAN(***list*,*frequency*) or **MEDIAN(***list*,*frequency*)

```
MEAN({1,2,3})
                2
MEAN({1,2,3},{4,
2,1})
     1.571428571
```

```
MEDIAN({1,2,3})
                2
MEDIAN({1,2,3},{
3,2,1})
              1.5
```

**SUM**

**SUM** (summation, **LIST MATH**, item 5) returns the sum of the elements in the specified list.

**SUM** *list*

```
{5,2,3}->L1
              {5 2 3}
SUM L1
                   10
```

**PROD**

**PROD** (product, **LIST MATH**, item 6) returns the product of the elements of the list.

**PROD** *list*

```
PROD {5,2,3}
                   30
```

**Sums and Products of Numeric Sequences**

You can combine **SUM** or **PROD** with **SEQ(** to obtain:

$$\sum_{x=lower}^{upper} expression(x) \qquad \prod_{x=lower}^{upper} expression(x)$$

To evaluate $\Sigma\, 2^{(N-1)}$ from N=1 to 4:

```
SUM SEQ(2^(n-1),
n,1,4,1)
                   15
```

# Chapter 9: Statistics

**This chapter describes the tools for analysing statistical data on the TI-80. These include entering lists of data, calculating statistical results, matching data to a model, and plotting data.**

**Chapter Contents**

# Getting Started: Building Height and City Size

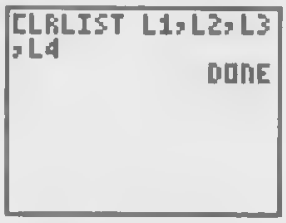
**Getting Started is a fast-paced introduction. Read the chapter for details.**

**Determine a linear equation to match the data below. Enter the data, and perform a linear regression. Then plot the data. Predict how many buildings of more than 12 stories you would expect to find in a city of 300 thousand people.**

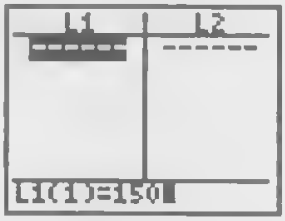
| Population in Thousands | Buildings > 12 Stories |
|---|---|
| 150 | 4 |
| 500 | 31 |
| 800 | 42 |
| 250 | 9 |
| 550 | 20 |
| 750 | 55 |

1. To clear any existing lists, press [STAT] **4** to copy **CLRLIST** to the Home screen.

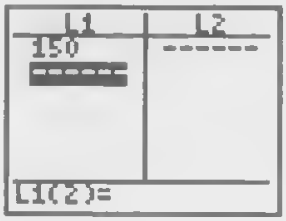

2. Press [2nd] [L1] [,] [2nd] [L2] [,] [2nd] [L3] [,] [2nd] [L4] [ENTER]. The message **DONE** is displayed.




3. Press [STAT] **1** to display the **STAT** list editor. Enter **150** to represent 150,000 for the population of the first city. As you type, the value is displayed on the bottom line.

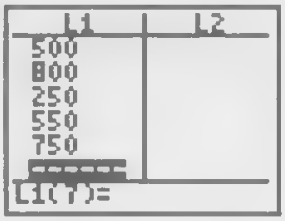



4. Press [ENTER]. The value is shown in the first element of **L1**, and the cursor moves to the second element in the same list.

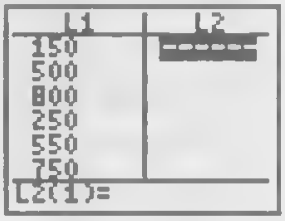


5. Enter the remaining elements of **L1**. Press:

   **500** [ENTER].
   **800** [ENTER].
   **250** [ENTER].
   **550** [ENTER].
   **750** [ENTER].

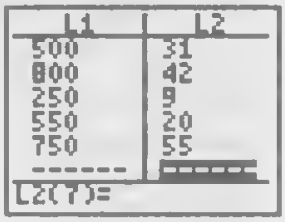



6. Press [▶] to move to the first element of list **L2**.



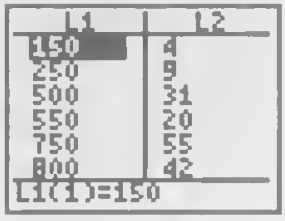

7. Enter the elements (number of buildings with more than 12 stories) of **L2**. Press:

   **4** [ENTER].
   **31** [ENTER].
   **42** [ENTER].
   **9** [ENTER].
   **20** [ENTER].
   **55** [ENTER].




8. You can sort the data according to city size. Press [2nd] [QUIT] [CLEAR] to return to a clear Home screen. Press [STAT] **2** to select **SORTA(**, which is copied to the Home screen. Press [2nd] [L1] to select the independent list and then press [,] [2nd] [L2] to select the dependent list. Press [)] [ENTER]. The message **DONE** is displayed. The lists have now been updated in memory.

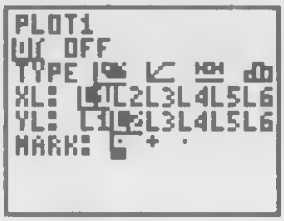

9. Press [STAT] **1** to display the sorted lists in the **STAT** list editor.



**After entering and sorting the data, define the STAT PLOTS and Window variables; then perform a linear regression (aX + b).**

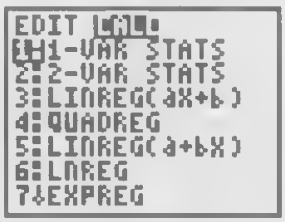
10. Press [2nd] [STAT PLOT] to display the **STAT PLOTS** screen.


11. Press **1** to display the **PLOT1** screen. Move the cursor to **ON**, if necessary, and press [ENTER] to turn **PLOT1** on. Leave **TYPE** as scatter plot (), **XL** (independent list) as **L1**, **YL** (dependent list) as **L2**, and **MARK** as □.



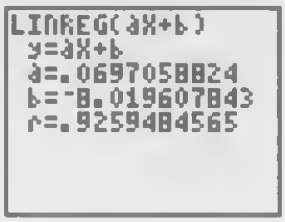

12. Press [WINDOW] to display the Window variables. Enter the following values.
**0** for **XMIN**
**1000** for **XMAX**
**100** for **XSCL**
**-15** for **YMIN**
**100** for **YMAX**
**10** for **YSCL**


13. Press [STAT] [▶] to display the **STAT CALC** menu.




14. Press **3** to select **LINREG(aX+b)**, which is copied to the Home screen. Press [2nd] [L1] [,] [2nd] [L2] [ENTER].

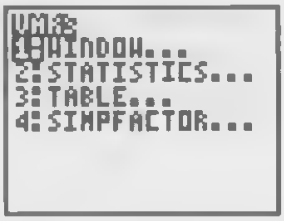
The least-squares linear regression is calculated; the display shows the values for **a** (slope), **b** (y-intercept), and **r** (correlation coefficient).



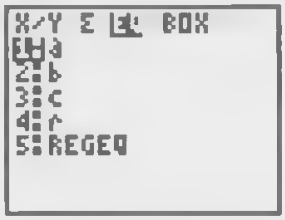
**Store the regression equation in the Y= list and graph it.**

15. In **FUNC** mode, press [Y=] to display the **Y=** editor. Press [CLEAR] to clear **Y1**, if necessary. Turn off all other functions, if necessary.

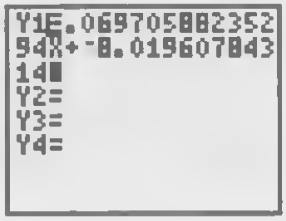

16. Press [VARS] to display the **VARS** menu.




17. Press **2** to select **STATISTICS...**, and press [▶] [▶] to display the **VARS EQ** menu.

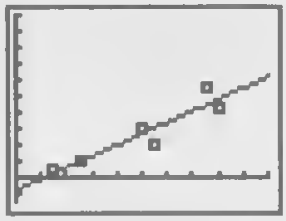



18. Press **5** to select **REGEQ**, which copies the linear regression to the **Y=** editor screen.

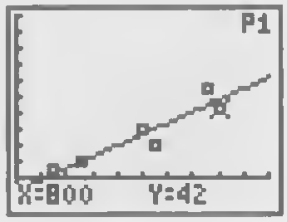



**Note:** Each time you calculate a regression, the regression equation (**REGEQ**) is updated.

19. Press [GRAPH]. The data points are plotted (□); then the regression line is drawn.



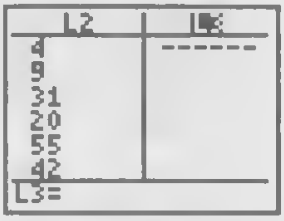
20. Press [TRACE] and then [▶] to trace the points in **PLOT1**, as indicated by **P1** in the upper right hand corner of the display.



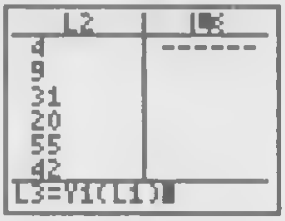

Press [▼] to move to **Y1**, and continue tracing the function.

**You can enter expressions to define lists in the STAT list editor. For example, you can now define predicted values and residuals (the differences between the observed values and the predicted values) for this problem.**

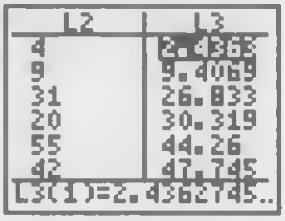
21. Press [STAT] **1** to display the **STAT** list editor. Press [▶] [▶] [▲] to move the cursor onto the name **L3**.



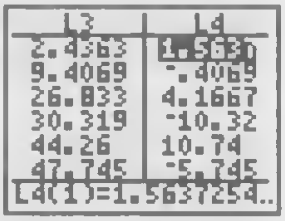

22. Press [2nd] [Y-VARS] **1** to select **Y1**, and then press [(] [2nd] [L1] [)]. This defines **L3** as the values predicted by the **LINREG** line.




23. Press [ENTER] to store the values in **L3**.




24. To store the residuals in **L4**, press [▶] [▲] to move the cursor onto the name **L4**. To enter **L4 = L2 - L3**, press [2nd] [L2] (the observed values) [−] [2nd] [L3] (the predicted values) [ENTER].

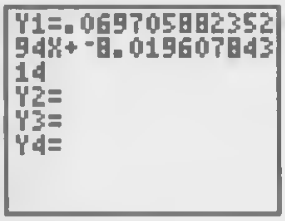



25. Press [2nd] [STAT PLOT]. Press **1** to select **PLOT1**. Move the cursor to **OFF**, and press [ENTER] to turn the plot off.


26. Press [2nd] [STAT PLOT] **2** to select **PLOT2.** Move the cursor to **ON** and press [ENTER] to turn the plot on, if necessary. Press [▼] [▼] [ENTER] to define **XL** as **L1**. Press [▼] [▶] [▶] [▶] [ENTER] to define **YL** as **L4**. Press [▼] [▶] [ENTER] to set **MARK** as +.

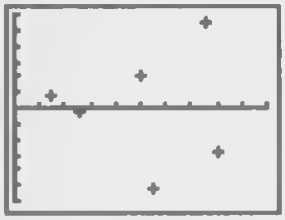
**Plot the residuals, and predict how many buildings of 12 or more stories there are in a city with a population of 300 thousand.**

27. Press [Y=] [◀] [ENTER] to turn off **Y1**.




28. Press [WINDOW], and change the Window variable values to best show the residuals. Use the minimum and maximum values of **L4** (-10.31862745 and 10.74019608) as guidelines for setting **YMIN** and **YMAX**.


29. Press [GRAPH] to plot the residuals. + marks each residual value.



30. Press [2nd] [QUIT] [CLEAR] to return to a clear Home screen. Press [2nd] [Y-VARS] **1** to select **Y1**. Then press [(] **300** [)] [ENTER].

    The value of **Y1** (the linear regression equation) for X=300 (which represents 300 thousand city population) is shown. Remember to round off the number mentally to an integer (13) to represent whole buildings.

# Setting Up a Statistical Analysis

**The data for statistical analyses is stored in lists. The TI-80 has six list variables (L1 to L6) that you can use in STAT calculations. Several types of statistical analysis are available.**

**Steps**

Follow these basic steps to perform statistical analyses.

1. Enter the statistical data in lists (pages 9-9 to 9-13).
2. Select the type of statistical calculations you want to do (pages 9-14 to 9-16), and specify the list names for the data.
3. Calculate the statistical variables, or match the data to a model (page 9-17).
4. Plot the data (pages 9-18 to 9-21).

# The STAT List Editor

**Pressing [STAT] accesses the STAT list editor and several instructions for use with lists (L1 to L6). The instructions are discussed on page 9-13.**

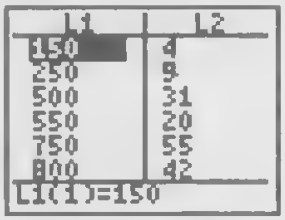
## STAT EDIT Menu

| EDIT CALC | |
|---|---|
| 1: EDIT... | Displays list editor. |
| 2: SORTA( | Sorts list in ascending order. |
| 3: SORTD( | Sorts list in descending order. |
| 4: CLRLIST | Deletes all elements of list. |

## Displaying the STAT List Editor

The **STAT** list editor gives you an easy environment in which to enter or edit lists. You can also create lists directly from the keyboard (Chapter 8), if you prefer.

To display the **STAT** list editor, press [STAT] and then press **1** or [ENTER] to select **EDIT...** from the **STAT EDIT** menu.




The top line of the **STAT** list editor displays the names of the lists (even if the list is empty). The centre portion displays up to six elements of two lists, showing the values of the elements (abbreviated to six digits if necessary).
The full value of the current element (indicated by the rectangular cursor) is shown on the bottom line.

## Entering List Elements in the STAT List Editor

To enter a list into the **STAT** list editor:

1. Display the **STAT** list editor.
2. Enter the first value in the list, and press [ENTER] or [▼]. The value is entered, and the rectangular cursor moves down to the next position.
3. Continue until you have entered all the data in the list.

Press [▶] and [◀] in the editor to move between lists.

**Note:** You may enter an expression, which is evaluated when you press [ENTER], [▼], or [▲].

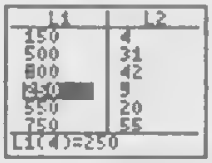
## Leaving the STAT List Editor

To leave the **STAT** list editor:

- Select another screen by pressing the appropriate key.
- Press [2nd] [QUIT] to return to the Home screen.

**The STAT list editor has two "contexts," viewing and editing. The current context determines the result of a key press. In both contexts, the full value of the highlighted element is displayed on the bottom line.**

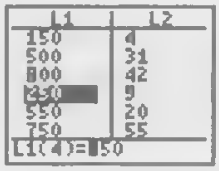
### Viewing Context




In the viewing context, you can move quickly from one list element to the next.

| | |
|---|---|
| ◄ or ► | Moves the rectangular cursor to the previous or next list. |
| ▼ or ▲ | Moves the rectangular cursor within the current column. On row 1, ▲ moves the cursor to the list name. |
| ENTER | Activates the edit cursor on the bottom line. |
| CLEAR | Clears the value on the bottom line. |
| Any entry character | Clears the value on the bottom line; copies the character to the bottom line. |
| 2nd [INS] | Inserts a list element (value is zero). |
| DEL | Deletes the current list element and closes the list. |

### Editing Context




In the editing context, an edit cursor is active on the bottom line and you can change the value of the current list element. You can also move the cursor onto the list name and edit the entire list at once.

| | |
|---|---|
| ◄ or ► | Moves the edit cursor within the value. |
| ▼ or ▲ | Stores the value on the bottom line to the list element; moves the rectangular cursor within the column. On row 1, ▲ moves the cursor to the list name. |
| ENTER | Stores the value on the bottom line to the list element; moves the rectangular cursor to the next element. |
| CLEAR | Clears the value on the bottom line. |
| Any entry character | Copies the character to the edit-cursor location on the bottom line. If it is the first character typed, the value on the bottom line is cleared. |
| 2nd [INS] | Activates the insert cursor. |
| DEL | Deletes a character. |

### Deleting a List

You can delete the contents of a list in several ways:

- With the **CLRLIST** instruction (page 9–13).
- Through the **MEMORY** menu (Chapter 12).
- In the **STAT** list editor, by pressing ▲ to move onto the list name and then pressing CLEAR ENTER.
- In the **STAT** list editor, by deleting each element.
- On a command line, by entering **0→DIM** *listname*.

### Editing a List Element

To edit a list element:

1. Display the **STAT** list editor.
2. Move the rectangular cursor to the element you want to change.
3. Press ENTER to switch to the editing context. Then:
   - Change the current value by inserting, deleting, or typing over digits.
   - Press an entry key, such as a number or letter, to begin an entry. This automatically clears the value.
   - Press CLEAR to clear the entire value and then enter a new value.

     **Note:** If you clear a value by mistake, you can immediately press ENTER to restore the value at the rectangular cursor.
4. Press ENTER to store the new value, and move to another element.

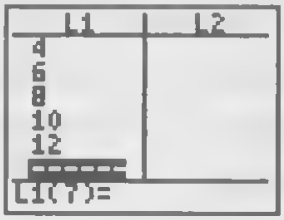
**Note:** You may enter an expression, which is evaluated when you press ENTER, ▼, or ▲.

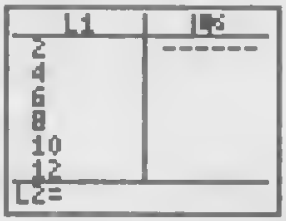
**You can enter or edit an entire list by moving the cursor to a list name on the top line of the STAT list editor and then pressing [ENTER]. The bottom line displays L*n*=L*n*×1, if there is already data in the list. Type any expression that returns a list, and press [ENTER]. The new list is displayed.**

### Entering an Entire List

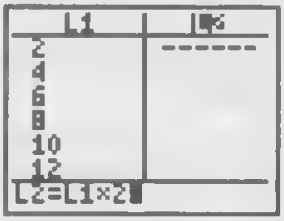
To enter an entire list:

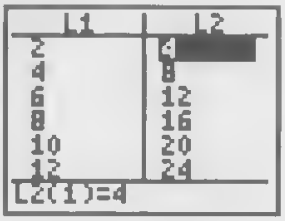
1. Press [STAT] [ENTER]. Enter several elements in **L1**.
2. Press [▲] and [▶] as many times as necessary to move the cursor to the list name **L2**.




3. Press [2nd] [L1] [×] **2**. This is the expression that will define the elements in **L2**.
4. Press [ENTER] to define and display **L2**.

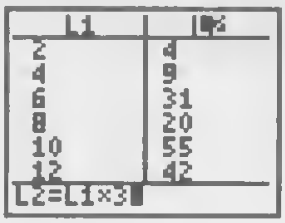


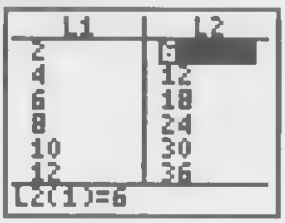




### Editing an Entire List

To replace an existing list:

1. Move the cursor to the list name **L2**. **L2=L2×1** is displayed.
2. Enter a new expression to replace the existing values in **L2**, [2nd] [L1] [×] **3**, for example. Then press [ENTER]. The values in **L2** are replaced, and the new values are displayed.



**Items 2 to 4 on the STAT EDIT menu—SORTA(, SORTD(, and CLRLIST—let you sort or clear list data. Pressing [STAT] displays these instructions, and selecting an item copies the name of the instruction to the Home screen. Note that SORTA( and SORTD( are the same as SORTA( and SORTD( on the LIST OPS menu (Chapter 8).**

**SORTA(**
**SORTD(**

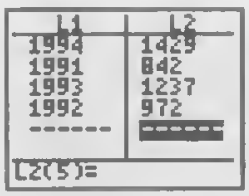
**SORTA(** (sort ascending, **STAT EDIT**, item 2) and **SORTD(** (sort descending, **STAT EDIT**, item 3) have two uses.

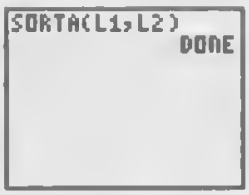
- With one list name, they sort the elements of an existing list and update the list in memory.
- With two to six list names, they sort the first list and then sort the remaining lists as dependent lists, placing their elements in the same order as their corresponding elements in the first list. This lets you sort two-variable data on X and keep the data pairs together.

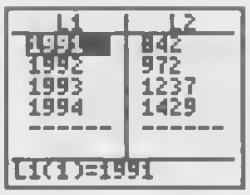
All of the lists to be sorted must be the same length. The sorted lists are updated in the memory.

**Note:** You can only reference a specific list once in these instructions.

**SORTA(***listname***)**
**SORTA(***keylistname,dependlistA,dependlistB*, . . .**)**
**SORTD(***listname***)**
**SORTD(***keylistname,dependlistA,dependlistB*, . . .**)**



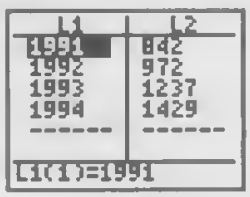



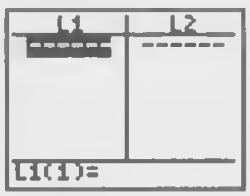

**CLRLIST**

**CLRLIST** (clear list, **STAT EDIT**, item 4) clears (deletes) the elements of one or more lists.

**CLRLIST** *listnameA,listnameB*, . . .







**Pressing [STAT] [▶] accesses the STAT CALC menu, where you select and perform statistical calculations. The TI-80 can analyse one-variable or two-variable statistics. Both can have associated frequency lists.**

## STAT CALC Menu

| EDIT CALC | |
|---|---|
| 1:1-VAR STATS | Calculates 1-variable statistics. |
| 2:2-VAR STATS | Calculates 2-variable statistics. |
| 3:LINREG(aX+b) | Matches data to linear model. |
| 4:QUADREG | Matches data to quadratic model. |
| 5:LINREG(a+bX) | Matches data to linear model. |
| 6:LNREG | Matches data to logarithmic model. |
| 7:EXPREG | Matches data to exponential model. |
| 8:PWRREG | Matches data to power model. |

## Selecting and Performing a Statistical Calculation

To select and perform statistical calculations:

1. Select a calculation type by pressing its corresponding number on the **STAT CALC** menu. The name of the calculation is copied to the Home screen.
2. Enter the name(s) of the list(s) to be used in the calculation. If you enter more than one list name, separate them with commas.
3. Press [ENTER] to perform the calculation and display the results.

## Frequency of Occurrence for Data Points

For all of the calculation types, you can include a list of data occurrences, or frequencies. These indicate how many times the corresponding data points or data pairs occur in the data set you are analysing.

For example, if **L1={15.5,12.1,9.8,14.7,15}** and **L2={1,4,1,3,3}**, then the instruction **1-VAR STATS L1,L2** would assume that 15.5 occurred one time, 12.1 occurred four times, 9.8 occurred one time, and so on.

Frequencies must be greater than or equal to zero. At least one frequency in the list must be greater than zero.

Noninteger frequencies are valid. This is useful in entering frequencies expressed as percentages or parts that add up to 1. Noninteger frequencies, however, may prevent the calculation of certain variables.

**These calculations return statistical results based on the list(s) you reference. If you reference a third list name as an argument for 2-VAR STATS or any of the regression models, the list is interpreted as the frequencies of occurrence for the data pairs in the first two lists.**

**1-VAR STATS**

**1-VAR STATS** (one-variable statistics, **STAT CALC**, item 1) analyses data with one measured variable and calculates statistical results as indicated on page 9-17.

If you reference two list names, the second list is interpreted as the frequency of occurrence for each data point in the first list.

**1-VAR STATS** *listname*
**1-VAR STATS** *Xlistname,freqlistname*


**2-VAR STATS**

**2-VAR STATS** (two-variable statistics, **STAT CALC**, item 2) analyses pairs of data between which there is a relationship. This option calculates statistical results as indicated on page 9-17.

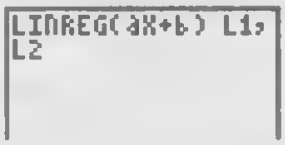
The first list you reference is the independent variable (X list). The second list is the dependent variable (Y list). If you reference a third list name, it is interpreted as the frequency of occurrence for each data pair in the first two lists.

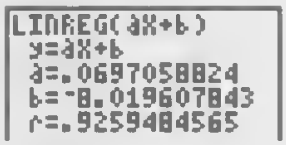
**2-VAR STATS** *Xlistname,Ylistname*
**2-VAR STATS** *Xlistname,Ylistname,freqlistname*

**LINREG (aX+b)**

**LINREG (aX+b)** (linear regression, **STAT CALC**, item 3) matches the data to the model y=ax+b using a least-squares match, and **x** and **y**. It displays **a** (slope), **b** (y-intercept), and **r** (correlation coefficient).

**LINREG (aX+b)** *Xlistname,Ylistname*
**LINREG (aX+b)** *Xlistname,Ylistname,freqlistname*



**QUADREG**

**QUADREG** (quadratic regression, **STAT CALC**, item 4) matches the data to the second-order polynomial $y=ax^2+bx+c$. It displays **a**, **b**, and **c**. For three data points the equation is a polynomial match; for four or more, it is a polynomial regression. At least three data points are required.

**QUADREG** *Xlistname,Ylistname*
**QUADREG** *Xlistname,Ylistname,freqlistname*

**LINREG (a+bX)**

**LINREG (a+bX)** (linear regression, **STAT CALC**, item 5) matches the data to the model equation y=a+bx using a least-squares match and **x** and **y**. It displays **a**, **b**, and **r** (correlation coefficient).

**LINREG (a+bX)** *Xlistname,Ylistname*
**LINREG (a+bX)** *Xlistname,Ylistname,freqlistname*

**LNREG**

**LNREG** (logarithmic regression, **STAT CALC**, item 6) matches the data to the model equation y=a+b ln(x) using a least-squares match and transformed values **LN(x)** and **y**. It displays **a**, **b**, and **r** (correlation coefficient).

**LNREG** *Xlistname,Ylistname*
**LNREG** *Xlistname,Ylistname,freqlistname*

**EXPREG**

**EXPREG** (exponential regression, **STAT CALC**, item 7) matches the data to the model equation $y=ab^x$ using a least-squares match and transformed values **x** and **LN(y)**. It displays **a**, **b**, and **r** (correlation coefficient).

**EXPREG** *Xlistname,Ylistname*
**EXPREG** *Xlistname,Ylistname,freqlistname*

**PWRREG**

**PWRREG** (power regression, **STAT CALC**, item 8) matches the data to the model equation $y=ax^b$ using a least-squares match and transformed values **LN(x)** and **LN(y)**. It displays **a**, **b**, and **r** (correlation coefficient).

**PWRREG** *Xlistname,Ylistname*
**PWRREG** *Xlistname,Ylistname,freqlistname*

Note: Calculations for $\bar{x}$, $\Sigma X$, $\Sigma X^2$, **SX**, $\sigma X$, $\bar{y}$, $\Sigma Y$, $\Sigma Y^2$, **SY**, $\sigma Y$, and $\Sigma XY$ are calculated using transformed values for **LNREG**, **EXPREG**, and **PWRREG**.

**The statistical variables are calculated as indicated below. Some are displayed when 1-VAR STATS or 2-VAR STATS are calculated. You can access these variables for use in expressions through the VARS STATISTICS... menus. If you edit a list, all statistical variables are cleared.**

| Variables | 1-VAR STATS | 2-VAR STATS | LIN, LN, EXP, PWR REGS | QUADREG | VARS Menu |
|---|---|---|---|---|---|
| $\bar{x}$ (mean of X values) | ✓ | ✓ | ✓ | | X/Y |
| **ΣX** (sum of X values) | ✓ | ✓ | ✓ | | Σ |
| **ΣX²** (sum of $X^2$ values) | ✓ | ✓ | ✓ | | Σ |
| **SX** (sample standard deviation of X) | ✓ | ✓ | ✓ | | X/Y |
| **σX** (population standard deviation of X) | ✓ | ✓ | ✓ | | X/Y |
| **n** (number of data points) | ✓ | ✓ | ✓ | ✓ | X/Y |
| $\bar{y}$ (mean of Y values) | | ✓ | ✓ | | X/Y |
| **ΣY** (sum of Y values) | | ✓ | ✓ | | Σ |
| **ΣY²** (sum of $Y^2$ values) | | ✓ | ✓ | | Σ |
| **SY** (sample standard deviation of Y) | | ✓ | ✓ | | X/Y |
| **σY** (population standard deviation of Y) | | ✓ | ✓ | | X/Y |
| **ΣXY** (sum of X * Y) | | ✓ | ✓ | | Σ |
| **MINX** (minimum of X values) | ✓ | ✓ | | | X/Y |
| **MAXX** (maximum of X values) | ✓ | ✓ | | | X/Y |
| **MINY** (minimum of Y values) | | ✓ | | | X/Y |
| **MAXY** (maximum of Y values) | | ✓ | | | X/Y |
| **Q1** (1st quartile) | ✓ | | | | BOX |
| **MED** (median) | ✓ | | | | BOX |
| **Q3** (3rd quartile) | ✓ | | | | BOX |
| **a, b** (regression/match coefficients) | | | ✓ | | EQ |
| **a, b, c** (quadratic coefficients) | | | | ✓ | EQ |
| **r** (correlation coefficient) | | ✓ | ✓ | | EQ |
| **REGEQ** (regression equation) | | | ✓ | ✓ | EQ |

### Q1 and Q3

The quartile **Q1** is the median of the ordinals to the left of **MED** (median). The quartile **Q3** is the median of the ordinals to the right of **MED**.

### Noninteger Frequencies

If a frequency list contains noninteger values, **SX** and **SY** are undefined. No values are displayed for them in the statistical results. **Q1**, **MED**, and **Q3** are also undefined if the frequency list contains noninteger values.

### Large Frequencies

If a frequency list contains a value larger than 99, **Q1**, **MED**, and **Q3** will not be calculated.

### Zero Frequencies

If the frequency for an element or data pair is zero, the element or data pair is ignored in the calculation.

# Statistical Plotting

**You can plot statistical data that you have entered in lists. The types of plot available include scatter plots, x-y lines, box and whisker plots, and histograms. You can define up to three plots at a time.**

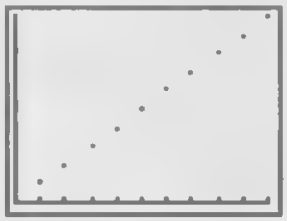
## Steps

To plot statistical data:

1. Enter the statistical data as lists (page 9-9 and Chapter 8).
2. Select the statistical calculations (pages 9-14 to 9-16), and calculate the statistical variables (page 9-17) or match the data to a model, if desired.
3. Select or deselect **Y=** equations, as appropriate (Chapter 4).
4. Define the statistical plot (page 9-20).
5. Turn the plot(s) on, if necessary (page 9-21).
6. Define the viewing window (page 9-21 and Chapter 4).
7. Display and explore the graph by pressing [GRAPH], [ZOOM], or [TRACE].

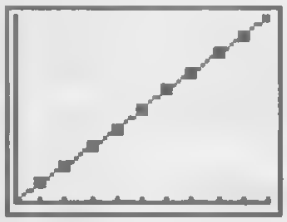
## Scatter Plot

(scatter plot) plots the data points from **XL** (X list) and **YL** (Y list) as coordinate pairs, showing each point as a box ( ▫ ), cross (+), or dot (•). **XL** and **YL** must be the same length. They can be the same list.

## XYLine

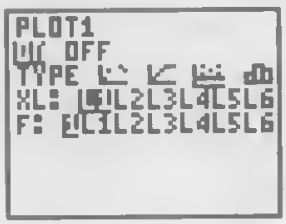
(XYLine) is a scatter plot in which the data points are plotted and connected in the order in which they appear in **XL** and **YL**. You may want to sort the lists with **SORTA(** or **SORTD(** before plotting.

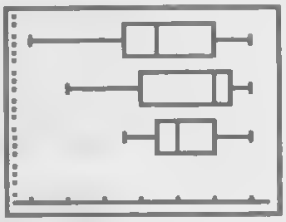



### Box Plot

(box plot) plots one-variable data. The whiskers on the plot extend from the minimum data point in the set (**MINX**) to the first quartile (**Q1**) and from the third quartile (**Q3**) to the maximum point (**MAXX**). The box is defined by **Q1**, the median (**MED**), and **Q3** (page 9-17).

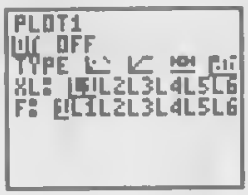
Box plots are plotted with respect to **XMIN** and **XMAX**, but ignore **YMIN** and **YMAX**. When two box plots are plotted, the first plots at the top of the screen and the second plots in the middle. When three are plotted, the first plots at the top, the second in the middle, and the third at the bottom.

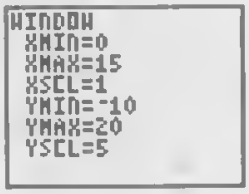


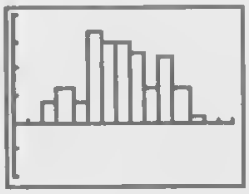



### Histogram

(histogram) plots one-variable data. **XSCL** determines the width of each bar, beginning at **XMIN**. **(XMAX – XMIN)/ XSCL** must be ≤ 31. A value occurring on the edge of a bar is counted in the bar to the right.



### Frequencies in Stat Plots

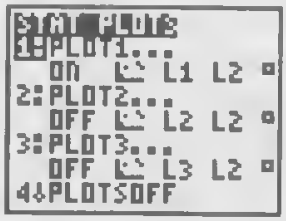
The frequency list specified for a statistical plot works just like the frequency lists specified for other statistical calculations (pages 9-14 to 9-16).

If you want to exclude an outlying data point from a plot, enter a zero for that value in the frequency list to prevent having to change the data lists.

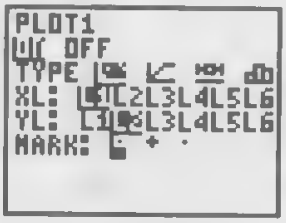
**Defining the Plots**

To define plots:

1. Press 2nd [STAT PLOT]. The **STAT PLOTS** screen shows the current plot definitions.




2. Select the plot to define (**PLOT1**, **PLOT2**, or **PLOT3**).
3. If you wish to immediately plot the statistical data, select **ON**. You can define a plot at any time and leave it **OFF**. The definition will be available in the future.




4. Select the type of plot. The options change appropriately:

| | | | |
|---|---|---|---|
| • (scatter plot): | **XL** | **YL** | **MARK** |
| • (XYLine): | **XL** | **YL** | **MARK** |
| • (box plot): | **XL** | **F** | |
| • (histogram): | **XL** | **F** | |

5. Depending on the type of plot, select the options:
   - **XL** (list of independent data)
   - **YL** (list of dependent data)
   - **F** (frequency; **1** is used if no list is specified)
   - **MARK** (□, +, or •)

### Turning Plots Off or On

**PLOTSOFF** and **PLOTSON** allow you to turn statistical plots off or on from the Home screen or a program. Used without *plot#*, they turn all plots off or all plots on. Used with *plot#*, they turn specific plots off or on.

**PLOTSOFF** or **PLOTSON**
**PLOTSOFF** *plot#,plot#, . . .*
**PLOTSON** *plot#,plot#, . . .*

For example, **PLOTSOFF** followed by **PLOTSON 3** turns all plots off and then turns **PLOT3** on.

```
PLOTSOFF
                DONE
PLOTSON 3
                DONE
```

### Defining the Viewing Window

Statistical plots are displayed on the current graph. You may define the viewing window by pressing [WINDOW] and then entering values for the Window variables.

### Tracing a Stat Plot

When you trace a scatter plot or XYLine, tracing begins at the first element in the list.

When you trace a box plot, tracing begins at **MED** (the median). Press [◀] to trace to **Q1** and **MINX**. Press [▶] to trace to **Q3** and **MAXX**.

When you trace a histogram, the cursor moves to the top centre of each column, starting at the first column.

When you press [▲] or [▼] to move to another plot or **Y=** function, tracing moves to the current or starting point on that plot.

# Statistical Analysis in a Program

**You can enter statistical data, calculate statistical results, and match data to models from a program.**

## Entering Stat Data

Enter the statistical data directly into lists (Chapter 8) in the program.

```
PROGRAM:STATS
:{25,36,42,54,64
}->L1
:{4,6,7,9,11}->L
2
:
```

## Statistical Calculations

To calculate statistical results or match data to a model from a program:

1. On a blank line in the program editor, select the type of calculation from the **STAT CALC** menu.
2. Enter the names of the lists, separated by commas, to be used in the calculation.

```
PROGRAM:STATS
:{25,36,42,54,64
}->L1
:{4,6,7,9,11}->L
2
:LINREG(aX+b) L1
,L2
:
```

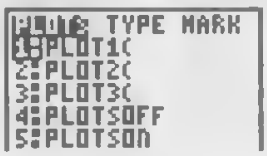
**Note:** To display a regression equation and coefficients from a program, the regression function must be the last statement in the program. If it is not the last statement the regression equation will be evaluated and stored, but the results will not be displayed

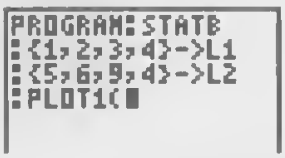
**To display a statistical plot, you may define the plot(s), then turn the plot(s) on, and then display the graph. If you do not define the plot, the current definitions are used.**

## Defining a Stat Plot In a Program

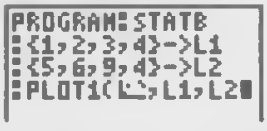
To define a statistical plot in a program:

1. Enter the data into list(s). On a blank line in the program editor, press [2nd] [STAT PLOT] to display the **PLOTS** menu.
2. Select the plot todbe defined. **PLOT1(**, **PLOT2(**, or **PLOT3(** is copied to the cursor location.

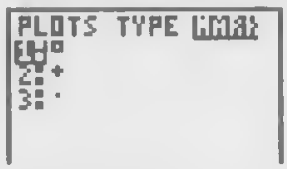


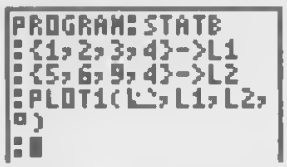




3. Press [2nd] [STAT PLOT] [▶] to display the **TYPE** menu. Select the type of plot. (scatter), (XYLine), (box), or (histogram) is copied to the cursor location.
4. Press [,]. Enter the list names, separated by commas.


5. (This step is for and only.) Press [,]. Press [2nd] [STAT PLOT] [▶] [▶] to display the **MARK** menu. Select the mark. □, +, or ·. is copied to the cursor location.
6. Press [)] and [ENTER] to complete the command line.



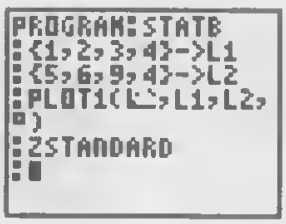



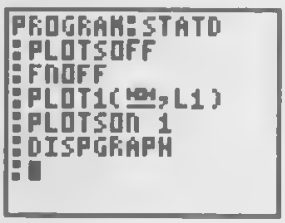

7. Press [2nd] [STAT PLOT] **5** to copy **PLOTSON** to the command line and the number of the plot (**1**, **2**, or **3**) to turn on. Press [ENTER] to complete the command line.


Note that **PLOTSOFF** in the example program ensures that all other plots are turned off.

**Displaying a Stat Plot from a Program**

To display a plot, use any of the Zoom instructions (Chapter 4), or use the **DISPGRAPH** instruction (Chapter 10).



# Chapter 10: Programming

**This chapter describes specific programming instructions and expiains how to enter and execute programs on the TI-80.**

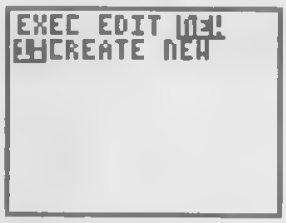
**Chapter Contents**

# Getting Started: Rolling a Dice

**Getting Started is a fast-paced introduction. Read the chapter for details.**

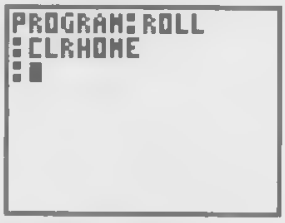
**A program is a set of commands that can be executed sequentially, as if they had been entered from the keyboard. Write a simple program to simulate the rolling of a single dice. It should prompt for the number of rolls and then store the results of the rolls in a list.**

1. Press [PRGM] [▶] [▶] to display the **PRGM NEW** menu.

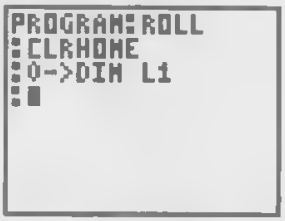



2. Press [ENTER] to select **CREATE NEW**. (The keyboard is now in **ALPHA-LOCK**.) Type **R O L L** as the name of the program, and press [ENTER]. You are now in the program editor. The **:** (colon) in the first column of the second line indicates that this is the beginning of a command line.

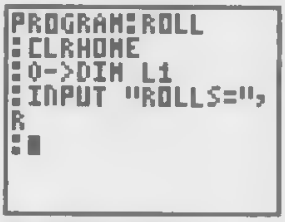

3. Press [PRGM] [▶] to access the **PRGM I/O** menu. Press **4**. **CLRHOME** is copied to the cursor location. Press [ENTER] to complete the instruction and move to the next line.




4. Press **0** [STO▸] [2nd] [LIST] **3** [2nd] [L1]. This sets the dimension of **L1** (the list where the results of the rolls will be stored) to 0. Press [ENTER] to complete the instruction and move to the next line.




5. Press [PRGM] [▶] **1** to copy **INPUT** to the cursor location. Press [2nd] [A-LOCK] ["] **ROLLS** [2nd] [TEST] **1** [ALPHA] ["] [,] [ALPHA] **R** to prompt the user to input the number of rolls. Press [ENTER] to complete the instruction.




6. Press [PRGM] **4** to copy **FOR(** to the cursor location. Press [ALPHA] **I** [,] **1** [,] [ALPHA] **R** [,] **1** [)]. Press [ENTER] to complete the instruction.

7. Press [MATH] [◀] to access the **MATH PRB** menu. Press **5** (to copy **RANDINT(** to the cursor location) and then **1** [,] **6** [)] [STO▶] [2nd] [L1] [(] [ALPHA] **I** [)] to generate random integers from 1 to 6, and store them into element **I** of **L1**. Press [ENTER] to complete the instruction.

```
PROGRAM:ROLL
:0->DIM L1
:INPUT "ROLLS=",
R
:FOR(I,1,R,1)
:RANDINT(1,6)->L
1(I)
:
```

8. Press [PRGM] [▶] **2** to select **DISP** (display), which is copied to the cursor location. Press [2nd] [L1] [(] [ALPHA] **I** [)]. This instruction displays the value of element **I** (the result of the last roll) in **L1**. Press [ENTER] to complete the instruction.

```
PROGRAM:ROLL
:INPUT "ROLLS=",
R
:FOR(I,1,R,1)
:RANDINT(1,6)->L
1(I)
:DISP L1(I)
:
```

9. Press [PRGM] **6** to select **PAUSE**, which is copied to the cursor location. This pauses the program after displaying the result of the last roll. Press [ENTER] to complete the instruction.

```
PROGRAM:ROLL
R
:FOR(I,1,R,1)
:RANDINT(1,6)->L
1(I)
:DISP L1(I)
:PAUSE
:
```

10. Press [PRGM] **5** to select **END**, which is copied to the cursor location. **END** identifies the end of the group of commands in the **FOR(** loop. Press [ENTER] to complete the instruction.

```
PROGRAM:ROLL
:FOR(I,1,R,1)
:RANDINT(1,6)->L
1(I)
:DISP L1(I)
:PAUSE
:END
:
```

11. Press [PRGM] [▶] **2** [2nd] [L1] to display the list of all the roll results. Press [ENTER] to complete the instruction.

12. Press [2nd] [QUIT] [CLEAR] [PRGM]. Move the cursor to the program name **ROLL**. Press [ENTER]. Press [ENTER] again from the Home screen to execute **PRGM_ROLL**.

```
PROGRAM:ROLL
:RANDINT(1,6)->L
1(I)
:DISP L1(I)
:PAUSE
:END
:DISP L1
:
```

# About TI-80 Programs

**Most features of the TI-80 are accessible from programs. Programs can access all variables and named items. The number of programs that the TI-80 can store is limited only by the memory available.**

## Notes about Programs

Programs are identified by names of up to seven characters, beginning with a letter.

A program consists of a series of program commands, which begin with a : (colon). A program command can be an expression (a command, usually a combination of variables, functions, and numeric values, that returns a value to **ANS**) or an instruction (a command, such as **GRIDON** or **PT-ON(**, that does not return a value to **ANS**).

The TI-80 checks for errors when you execute the program, not as you enter or edit the program.

Variables and lists saved in the memory are global; i.e., they can be accessed from all programs. Storing a new value to a variable or list in a program changes the value in the memory during program execution.

As calculations are made in programs, the TI-80 updates **ANS**, just as it would if the calculations were being carried out on the Home screen. Programs do not update Last Entry as each command is executed.

## "Breaking" a Program

Pressing [ON] stops program execution. When you press [ON] during program execution, the **ERR: BREAK** menu is displayed.

- To go to where the interruption occurred, select **GOTO**.
- To return to the Home screen, select **QUIT**.

## Memory Management and Erasing Programs

The size of programs you can store is limited only by the memory available. To access the **MEMORY** menu, press [2nd] [MEM] from the Home screen. Memory status is displayed on the **MEMORY CHECK RAM...** screen. To increase available memory, you can delete items, including other programs, from the **MEMORY DELETE...** screen (Chapter 12).

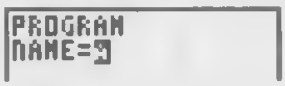
**Note:** Each token in a program takes 1 byte. For example, **SIN1.23** takes 5 bytes.

**Access the program editor by pressing [PRGM]. Then either choose to create a new program or edit an existing program. In general, anything that can be executed from the Home screen can be included in a program. A program command always begins with a colon (:).**

### Creating a New Program

To create a new program:

1. Press [PRGM] [▶] [▶] to display the **PRGM NEW** menu. Press [ENTER] to select **CREATE NEW**.




2. The keyboard is in **ALPHA-LOCK**. Enter the name you want for the program, followed by [ENTER]. The name may consist of between one and seven characters (**A**-**Z**, θ, 0-9) and must begin with a letter.
3. Enter the program commands.

### Entering Program Commands

A colon (:) indicates the beginning of each program command. To enter more than one command on a line, separate them with a colon, just as you would on the Home screen. Press [ENTER] to indicate the end of a command line.

When a command is longer than one line on the screen, it wraps to the next line. [2nd] [◀] and [2nd] [▶] move the cursor to the beginning and end of a command line.

In the program editor, if you press a key that accesses a menu, the menu screen temporarily replaces the program edit screen. When you make a selection or press [CLEAR], you are returned to the program editor.

### Leaving the Program Editor

When you finish entering or editing a program, press [2nd] [QUIT] to return to the Home screen. You must be on the Home screen to execute a program.

### Executing a Program

To execute a program:

1. From a blank line on the Home screen, press [PRGM] to display the **PRGM EXEC** menu. The names of all existing programs are listed, in alphabetical order.
2. Select a program. **PRGM_** and the program name are copied to the Home screen; for example, **PRGM_ROLL**.
3. Press [ENTER] to begin program execution. While the program is being executed, the busy indicator is displayed.

**The program editor also lets you edit an existing program. As you edit, you can enter commands just as you did when you created the program.**

### Editing a Program

To edit a program:

1. Press [PRGM] [▸] to display the **PRGM EDIT** menu.


2. Select the name of an existing program. The program editor and the commands in that program are displayed.
3. Edit the program by changing, inserting, or deleting commands, as needed.

### Changing Instructions

Move the cursor to the command you want to change.

- Position the cursor, and then make the changes by typing over the command or using [DEL] or [2nd] [INS].
- Press [CLEAR] to clear (blank) all program commands on the command line (The initial colon is not deleted.), and then enter a new program command.

### Inserting a New Command Line

To insert a new command line, position the cursor where you want to insert the new line, press [2nd] [INS] to put the TI-80 in insert mode, and then press [ENTER].

### Deleting a Command Line

To delete a command line, press [CLEAR] to clear the line, and then press [DEL] to delete the colon.

**Note:** All programs end with a blank command line; the colon on this line cannot be deleted.

# PRGM CTL (Control) Instructions

**PRGM CTL (program control) instructions can only be accessed from within the program editor. They direct the flow within a program being executed, making it easy to repeat or skip commands during program execution. While the program editor is displayed, press [PRGM]. The selected menu item is copied to the cursor location.**

## PRGM CTL Menu

| CTL I/O EXEC | |
|---|---|
| 1:IF | Creates a conditional test. |
| 2:THEN | Used with **IF**. |
| 3:ELSE | Used with **IF-THEN**. |
| 4:FOR( | Creates incrementing loop. |
| 5:END | Signifies end of loop, **IF-THEN**, or **ELSE**. |
| 6:PAUSE | Pauses program execution. |
| 7:LBL | Defines a label. |
| 8:GOTO | Goes to a label. |
| 9:PRGM_ | Executes a program as a subroutine. |
| 0:RETURN | Returns from a subroutine. |
| A:STOP | Stops program execution. |

## Controlling Program Flow

Program control instructions tell the TI-80 which command to execute next in a program. **IF** checks a condition that you define, to determine what command to execute next. The condition frequently uses relational tests (Chapter 2), such as **IF A<7:A+1→A**.

## IF

**IF** (**PRGM CTL**, item 1) is used for testing and branching. If the condition is false (zero), the command immediately following **IF** is skipped. If the condition is true (nonzero), that command is executed. **IF** instructions can be nested.

**:IF** *condition*
*:command if true*
*:command*

```
PROGRAM:COUNT
:0→A
:LBL 2
:A+1→A
:DISP "A IS",A
:IF A≥2
:STOP
:GOTO 2
```

```
PRGM_COUNT
A IS
                1
A IS
                2
            DONE
```

**IF-THEN**
**END**

**THEN** (**PRGM CTL**, item 2) following an **IF** executes a group of commands if the condition is true (nonzero).

**END** (**PRGM CTL**, item, 5) identifies the end of the group.

**:IF** *condition*
**:THEN**
:*command if true*
:...
**:END**
:*command*

```
PROGRAM:TEST
:1->X:10->Y
:IF X<10
:THEN
:2X+3->X
:2Y-3->Y
:END
:DISP X,Y
```

```
PRGM_TEST
                5
               17
             DONE
```

**IF-THEN-ELSE**
**END**

**ELSE** (**PRGM CTL**, item 3) following **IF-THEN** executes a group of commands if the condition is false (zero).

**END** (**PRGM CTL**, item 5) identifies the end of the group.

**:IF** *condition*
**:THEN**
:*command if true*
:...
**:ELSE**
:*command if false*
:...
**:END**
:*command*

```
PROGRAM:TESTELS
:INPUT "X=",X
:IF X<0
:THEN
:X²->Y
:ELSE
:X->Y
:END
:DISP {X,Y}
```

```
PRGM_TESTELS
X=5
              {5 5}
               DONE
X=-5
            {-5 25}
               DONE
```

**FOR(**
**END**

**FOR(** (**PRGM CTL**, item 4) is used for looping and incrementing. It increments the variable from the start value to the finish value, by the specified increment. The finish value is a maximum or minimum value that is not to be exceeded. The increment is optional (if not specified, 1 is used) and can be negative (finish value < start value).

**END** identifies the end of the loop. **FOR(** loops can be nested.

:**FOR(***variable,begin,end,increment*)
:*command while end not exceeded*
:...
:**END**
:*command*

```
PROGRAM:SQUARE
:FOR(A,0,8,2)
:DISP A²
:END
```

```
PRGM_SQUARE
                0
                4
               16
               36
               64
             DONE
```

**END**

**END** (**PRGM CTL**, item 5) identifies the end of a group of commands. Each **FOR(** and each **IF-THEN** or **IF-THEN-ELSE** group must have an **END** at the "bottom."

**PAUSE**

**PAUSE** (**PRGM CTL**, item 6) suspends execution of the program so that you can see answers or graphs. During the pause, the dotted pause indicator is displayed. When **DISP** or **DISPGRAPH** is executed, the appropriate screen is displayed. Press [ENTER] to resume program execution.

```
PROGRAM:PAUSE
:DISP "A=",A
:PAUSE
:DISPGRAPH
:PAUSE
:DISP
```

**LBL**
**GOTO**

**LBL** (label) and **GOTO** (go to) are used together for branching.

**LBL** (**PRGM CTL**, item 7) gives a label to a command line in a program. The label is one character (A-Z, 0-9, or θ).

**LBL** *label*

**GOTO** (**PRGM CTL**, item 8) causes the program to branch to the command line with the same label.

**GOTO** *label*

```
PROGRAM:CUBE
:LBL C
:INPUT A
:IF A≥100
:STOP
:DISP A³
:PAUSE
:GOTO C
```

```
PRGM_CUBE
?2
                  8
?3
                 27
?105
               DONE
```

**PRGM_**

**PRGM_** (**PRGM CTL**, item 9) calls (executes) other programs as subroutines (page 10-14). When you select **PRGM_**, it is copied to the cursor location. You may type the letters of an existing program name. You may also enter the name of a program you have not yet created.

When encountered, the command executes the specified program and then returns to the calling program. Execution continues with the command following **PRGM**_*programname*.

**PRGM**_*programname*

**RETURN**

**RETURN** (**PRGM CTL**, item 0) quits the subroutine and returns to the calling program (page 10-14), even if it is encountered within nested loops. (Any loops are ended.) There is an implied **RETURN** at the end of any program called as a subroutine. Within the main program, **RETURN** stops program execution and returns to the Home screen.

**STOP**

**STOP** (**PRGM CTL**, item A) stops execution of a program and returns you to the Home screen. **STOP** is optional at the end of a program. There is an implied stop at the end of the main program that is being executed.

**The PRGM I/O (program input/output) instructions can only be accessed from the program editor. They control input to and output from a program, allowing you to enter values and display answers during program execution. While the program editor is displayed, press [PRGM]. The selected menu item is copied to the cursor location.**

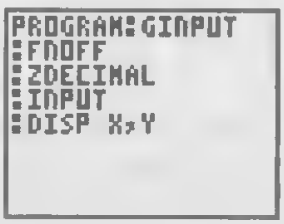
### PRGM I/O Menu

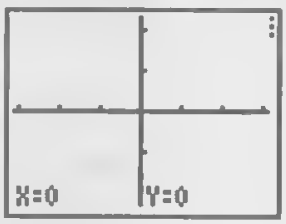
| CTL I/O EXEC | |
|---|---|
| 1: INPUT | Enters a value or displays the current graph. |
| 2: DISP | Displays text, value, or the Home screen. |
| 3: DISPGRAPH | Displays the current graph. |
| 4: CLRHOME | Clears the Home screen. |

### Displaying a Graph with INPUT

**INPUT** without a variable displays the current graph. You can move the free-moving cursor, which updates **X** and **Y**. The dotted pause indicator is displayed. Press [ENTER] to resume program execution.

For example, **INPUT** during program execution displays:



### Storing a Variable Value with INPUT

**INPUT** with a variable displays a **?** prompt during program execution. The value may be a real number, a list, or **Y=** function. Enter a value (which can be an expression or a list), and press [ENTER]. The value is evaluated and stored to the variable, and the program resumes execution.

**INPUT** *variable*
**INPUT** *listname*
**INPUT** *Yname*

```
PROGRAM:HINPUT
:INPUT A
:INPUT L1
:INPUT Y1
:DISP Y1(A)
:DISP Y1(L1)
```

```
PRGM_HINPUT
?2
?{1,2,3}
?"2X+5"
                9
        {7 9 11}
            DONE
```

You can enter a string of up to 16 characters up to the prompt, for the value to be entered. After the prompt, enter a value (which can be an expression or a list), and press [ENTER]. The value is stored to the variable, and the program resumes execution.

**INPUT** "*string*",*variable*
**INPUT** "*string*",*listname*
**INPUT** "*string*",*Yname*

```
PROGRAM:HINPUT
:INPUT "A=",A
:INPUT "L1=",L1
:INPUT "Y1=",Y1
:DISP Y1(A)
:DISP Y1(L1)
```

```
PRGM_HINPUT
A=2
L1={1,2,3}
Y1="2X+5"
                9
        {7 9 11}
            DONE
```

**Note:** When you input lists and expressions during program execution, you must include the braces ({}) around the list values and quotation marks (") around expressions.

### Displaying the Home Screen

**DISP** (display, **PRGM I/O**, item 2) with no value displays the Home screen. To view the Home screen during program execution, follow the **DISP** instruction with a **PAUSE**.

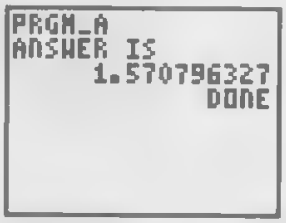
### Displaying Values and Messages

**DISP** (display, **PRGM I/O**, item 2) with one or more values displays the value of each.

**DISP** *value*
**DISP** *valueA*,*valueB*

- If *value* is a variable, the current value is displayed.
- If *value* is an expression, it is evaluated and then displayed, according to the current mode settings, on the right hand side of the following line.
- If *value* is text within " marks, it is displayed on the left hand side of the current display line.

For example, **DISP "ANSWER IS",π/2** displays:


If **PAUSE** is encountered after **DISP**, the program halts temporarily so that you can examine the screen. Press [ENTER] to resume program execution.

**Note:** A statement that results in a value will display without using **DISP**, if it is the last statement (other than **STOP**, **END**, and **PAUSE**) in the program

### DISPGRAPH

**DISPGRAPH** (display graph, **PRGM I/O**, item 3) displays the current graph. If **PAUSE** is encountered after **DISPGRAPH**, the program halts temporarily so you can examine the screen. Press [ENTER] to resume execution.

### CLRHOME

**CLRHOME** (clear Home screen, **PRGM I/O**, item 4) clears the Home screen during execution and places the cursor in the top left hand corner; however, program execution does not pause unless **PAUSE** is encountered.

**On the TI-80, any program can be called from another program, as a subroutine. Enter the name of the program to be used as a subroutine on a line of its own.**

## Calling a Program from Another Program

To call one program from another, begin on a blank line in the program editor and do one of the following.

- Press PRGM ◀ to display the **PRGM EXEC** menu, and select the name of the program. **PRGM_** and the name are copied to the cursor location, or.
- Select **PRGM_** from the **PRGM CTL** menu and then type the letters of the *programname*.

**PRGM**_*programname*

When this instruction is encountered during program execution, the next command that the program executes is the first command in the second program. Execution continues with the subsequent command in the first program when it encounters either a **RETURN** instruction or when the implied **RETURN** at the end of the called program is encountered.

```
PROGRAM:VOLCYL
:INPUT "D=",D
:INPUT "H=",H
:PRGM_AREACIR
:A×H->V
:DISP V
```

```
PROGRAM:AREACIR
:D/2->R
:π×R²->A
:RETURN
```

```
PRGM_VOLCYL
D=4
H=5
        62.83185307
              DONE
```

## Notes about Calling Programs

Variables are global.

A label used with **GOTO** and **LBL** is local to the program in which it is located. A label in one program is not "known" by another program. You cannot use **GOTO** to branch to a label in another program.

**RETURN** exits a subroutine and returns to the calling program, even if encountered within nested loops.

# Chapter 11: Applications

**This chapter contains application examples that incorporate the TI-80 features described in the preceding chapters. Several of the examples use programs.**

**Chapter Contents**

# Probability Experiments: Coins, Dice, and Spinners

**The RANDINT( (random integer) function can be used for probability experiments. RANDINT( takes two arguments that define a set of integers from which to draw for the probability experiments below.**

**Problem**

Using **RANDINT(** from the **MATH PRB** menu, devise probability experiments that employ the toss of a coin, the roll of two dice, and the spin of a wheel.

**Procedure**

For the coin tossing experiment, enter **RANDINT(0,1)** from the Home screen. 0 = tails and 1 = heads. Press [ENTER] repeatedly to generate the coin tosses.

```
RANDINT(0,1)
                1
                0
                1
                0
                1
                1
```

You can also write a simple program to display "heads" or "tails" for each coin toss.

```
PROGRAM:COIN
:RANDINT(0,1)->R
:IF R=0
:DISP "TAILS"
:IF R=1
:DISP "HEADS"
```

```
PRGM_COIN
TAILS
                DONE
HEADS
                DONE
HEADS
                DONE
```

You can simulate the rolling of two dice by adding together the result from each die after a roll. On the Home screen, enter **RANDINT(1,6)+RANDINT(1,6)** and press [ENTER] repeatedly.

```
RANDINT(1,6)+RAN
DINT(1,6)
               10
                4
                5
                9
                7
```

You can simulate spinning a wheel with the numbers 1 to 100 using the **RANDINT(** function. Enter **RANDINT(1,100)** on the Home screen and press [ENTER] repeatedly.

```
RANDINT(1,100)
               76
               98
               40
               25
                9
               66
```

# The Unit Circle and Trigonometric Curves

**You can use the parametric graphing feature of the TI-80 to show the relationship between the unit circle and any trigonometric curve.**

**Problem**

Graph the unit circle and the sine curve to demonstrate graphically the relationship between them.

Any function that can be plotted in function graphing can be plotted in parametric graphing by defining the X component as T and the Y component as F(T).

**Procedure**

Following this procedure to solve the problem.

1. Press [MODE] and select **RADIAN**, **PARAM**, and **SIMUL**.
2. Press [WINDOW] and set the Window variables.

| | | |
|---|---|---|
| **TMIN = 0** | **XMIN = -2** | **YMIN = -3** |
| **TMAX = 2π** | **XMAX = 2π** | **YMAX = 3** |
| **TSTEP = .1** | **XSCL = π/2** | **YSCL = 1** |

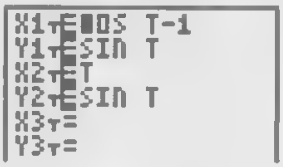
3. Press [Y=] and enter the expressions to define the unit circle centred at (-1,0).

**X1T=COS T-1** **Y1T=SIN T**

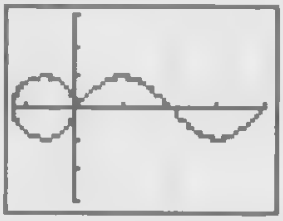
Enter the expressions to define the sine curve.

**X2T=T** **Y2T=SIN T**

Turn off all other functions.




4. Press [GRAPH] to see the **SIN** function "unwrap" from the unit circle.



**Note:** The "unwrapping" can be generalised. Replace **SIN T** in **Y2T** with any other trig function to "unwrap" that function.

# Program: Newton's Numerical Solve Routine

**This program uses the Newton-Raphson method to find the roots (zeros) of a function numerically.**

**Problem**

Find the roots of $e^X - 3X$.

**Program**

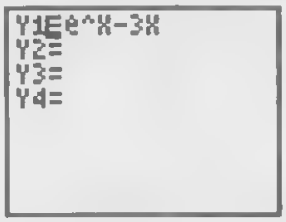
This program uses the Newton-Raphson method to find a root of **Y1** based on an initial guess. The program prompts for the initial guess. One way to make this initial guess is to graph and trace the function, and then enter X as the guess.

```
PROGRAM:NEWTON
:INPUT "INITIAL X=",X          Input initial guess.
:LBL N                         Begin loop.
:X-Y1/NDERIV(Y1,X,X)→R         Calculate new root.
:DISP R
:PAUSE
:IF ABS (X-R)≤ABS (X/1E10)     Test for convergence.
:STOP
:R→X                           Estimate with new root.
:GOTO N
```

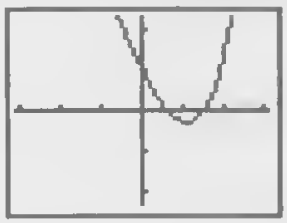
**Procedure**

Follow this procedure to solve the problem.

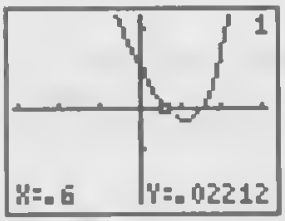
1. Press MODE and select **FUNC**.
2. Enter the program.
3. Press Y=. Enter the expression, **e^X−3X**, to define **Y1**.




4. Graph the function using **ZDECIMAL** from the **ZOOM** menu.



5. Press [TRACE] and move the cursor close to the left root. The variables X and Y are updated as you move the cursor.




6. From a blank line on the Home screen, execute the program **NEWTON**.

7. Enter **X** as the initial guess, and press [ENTER] repeatedly. The program stops when the relative difference between the new root estimate and the previous root estimate is less than X E − 10.

```
prgm_NEWTON
INITIAL X=X
      .6187784694
      .6190612219
      .6190612867
      .6190612867
                DONE
```

8. When program execution is complete, evaluate the function at the estimated root.

```
Y1(X)
                  0
```

Repeat the steps in this procedure to find the other root.

# Program: Numerical Integration

**This program uses Simpson's method to estimate the definite integral of a function.**

**Problem**

Estimate the definite integral of

$$\int_0^{1.5} (6-6x^5)dx$$

and graph the area of the integral.

**Program**

The program estimates the definite integral for **Y1** using Simpson's method. It prompts for the lower and upper limits of integration and the number of divisions.

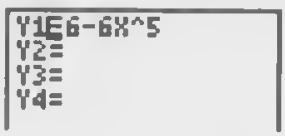
```
PROGRAM:SIMPSON
:INPUT "LOWER LIMIT=",A      Input lower limit.
:INPUT "UPPER LIMIT=",B      Input upper limit.
:INPUT "N DIVISIONS=",D      Input # of divisions.
:0→S                         Initialise sum variable.
:(B-A)/(2×D)→W               Calculate division width.
:FOR(J,1,D,1)                Begin calculation loop.
:A+2(J-1)W→L                 Calculate left point.
:A+2JW→R                     Calculate right point.
:(L+R)/2→M                   Calculate midpoint.
:W(Y1(L)+4Y1(M)+Y1(R))       Calculate division sum and
/3+S→S                       add to total.
:END
:DISP "AREA="                Display results.
:DISP S
```

**Procedure**

Follow this procedure to solve the problem.

1. Enter the program.

2. Press [Y=] and enter the function in **Y1**. Turn any other functions off.



3. Set the Window variables.

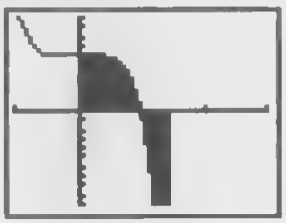
```
WINDOW
 XMIN=-1
 XMAX=3
 XSCL=1
 YMIN=-10
 YMAX=10
 YSCL=1
```

4. Execute the program **SIMPSON** from a clear Home screen, entering the limits and divisions as you are prompted.

```
PRGM_SIMPSON
LOWER LIMIT=0
UPPER LIMIT=1.5
N DIVISIONS=30
AREA=
     -2.390626758
             DONE
```

5. You can display the calculated area graphically, using **SHADE(** from a clear Home screen.

```
SHADE(MIN(0,Y1),
MAX(0,Y1),1,A,B)
```



# Program: Window Variables Store and Recall

**The program below lets you store the values for the current Window variables, and it lets you display a graph using previously stored values. It also demonstrates a method for including menus in a program.**

**Program**

```
PROGRAM:WINMEM
:CLRHOME
:DISP "WINDOW MEMORY"
:DISP "1:STORE WINDOW"
:DISP "2:RECALL WINDOW"
:DISP "3:QUIT"
:DISP " "
:DISP "ENTER 1,2, OR 3"
:INPUT M
:IF M=1
:GOTO S
:IF M=2
:GOTO R
:GOTO Q
:LBL S
:XMIN→A
:XMAX→B
:XSCL→C
:YMIN→D
:YMAX→E
:YSCL→F
:DISP "WINDOW STORED"
:GOTO Q
:LBL R
:A→XMIN
:B→XMAX
:C→XSCL
:D→YMIN
:E→YMAX
:F→YSCL
:DISPGRAPH
:GOTO Q
:LBL Q
:STOP
```

Present menu of choices.

Evaluate menu selection.

Store current graph Window variable values.

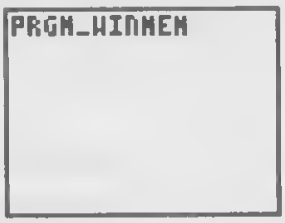
Display graph with previously stored variable values.

Quit program.

**Procedure**

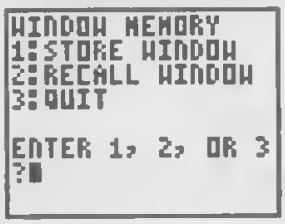
Follow this procedure to see how the program works.

1. Execute the program **WINMEM** from a blank Home screen.




The program prompts with three options.

- Store the Window variables you are currently using.
- View a graph using a previously stored set of Window variables.
- Quit the program.




2. Press **1**, **2**, or **3** and then ENTER to respond to the prompts.

The Window values are stored in variables A, B, C, D, E, and F.

# Graphing the Inverse of a Function

**You can use the parametric graphing feature of the TI-80 to graph the inverse relation of any function, by defining the function in X1T and Y1T and its inverse in X2T and Y2T.**

**Problem**

The function $Y=.2X^3-2X+6$ can be expressed in parametric form as $X_T=T$ and $Y_T=.2T^3-2T+6$.

The inverse relation of the function can be expressed in parametric form as $X_T=F(T)$ and $Y_T=T$. For example, $Y=.2X^3-2X+6$ would be expressed as $X_T=.2T^3-2T+6$ and $Y_T=T$.

Graph the function $Y=.2X^3-2X+6$ and its inverse.

**Procedure**

Follow this procedure to solve the problem.

1. Select **PARAM**, **CONNECTED**, and **SIMUL** modes.
2. Change the Window variable values.

| | | |
|---|---|---|
| **TMIN=-10** | **XMIN=-15** | **YMIN=-9** |
| **TMAX=10** | **XMAX=15** | **YMAX=9** |
| **TSTEP=.4** | **XSCL=1** | **YSCL=5** |

3. Enter the expressions to define the function in parametric form.

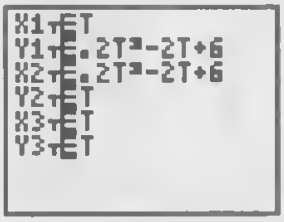
   **X1T=T**
   **Y1T=.2T^3–2T+6**

4. Enter the expressions to define the inverse in parametric form.

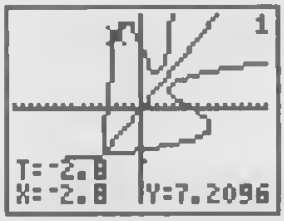
   **X2T=.2T^3–2T+6**
   **Y2T=T**

5. Enter the expressions to define the line Y=X, about which the graph of the function and the graph of its inverse are symmetric. That is, the reflection of the graph of the function through the line Y=X produces the graph of its inverse.

**X3T=T**
**Y3T=T**




6. Press [GRAPH] to plot the graph. Press [TRACE], and then press [▶] several times (until the cursor appears). Next, press [▼] and [▲] several times to move the cursor from a point on the relation to the reflected point and back again.





**Note:** The expressions to define the inverse can be generalised.

**X1T=Y2T**
**Y1T=X2T**

# Graphing a Piece-by-Piece Function

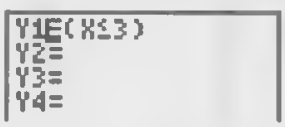
**The test functions of the TI-80 can be used to build piece-by-piece defined functions.**

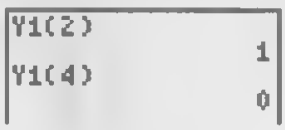
**Problem**

Define and graph this piece-by-piece defined function.

$$f(x)=\begin{cases} x^2, & \text{for } x\le 3 \\ 1.5x+1, & \text{for } 3<x<5 \\ 6-x, & \text{for } x\ge 5 \end{cases}$$

The **TEST** functions, which return 1 if true and 0 if false, can be used to build piece-by-piece defined functions. For example, when x is 4 (x≤3) is false and will return 0.

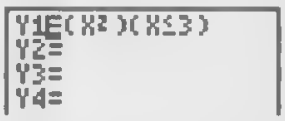


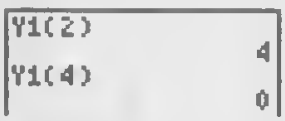




**Procedure**

Follow this procedure to solve the problem.

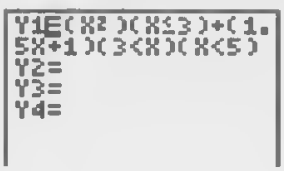
1. Press [MODE] and select **FUNC**.
2. Enter the first piece of the function in the **Y=** editor. This piece is $f(x)=x^2$ for x≤3. It is entered as **$(x^2)(x\le 3)$**. **Y1** is then equivalent to $x^2\times 1$ for x≤3 and $x^2\times 0$ for x>3.

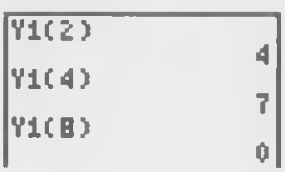


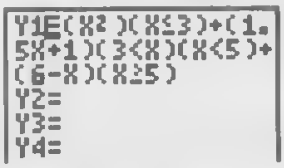

3. Add the second piece of the function in the **Y=** editor. This piece is f(x)=1.5x+1 for 3<x<5. It is entered as **(1.5x+1)(3< x)(x<5)**.

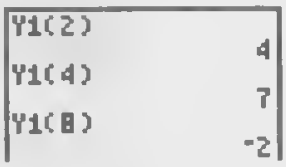
   When x is less than 3, the test (3< x) returns 0, and the test (x<5) returns 1. In this case, the second piece of the function is equivalent to $(1.5x+1)\times 0\times 1$, which is 0. Only when both of these tests are true will the second piece of this function be anything but 0.



4. Add the third piece of the function to **Y1**. This piece is f(x)=6−x for x≥5. It is entered as **(6−x) (x≥5)**. When x is greater than or equal to 5, the test (x≥5) returns 1. The third piece of the function is then equivalent to (6−x)×1. When X is less than 5, the test (x≥5) returns 0. The third piece of the function is then equivalent to (6−x)×0.



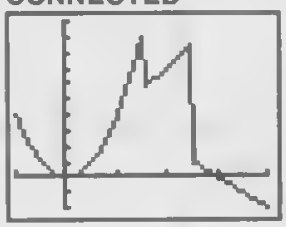




5. Enter these Window variable values.

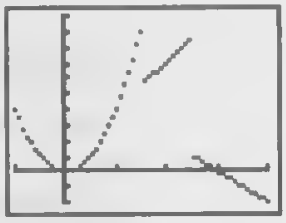
| | |
|---|---|
| **XMIN=-2** | **YMIN=-2** |
| **XMAX=8** | **YMAX=10** |
| **XSCL=2** | **YSCL=1** |

6. First graph the piece-by-piece function in **CONNECTED** and then **DOT** mode. Select **CONNECTED** on the **MODE** screen, and then press GRAPH. Then select **DOT** on the **MODE** screen, and press GRAPH.

**CONNECTED**



**DOT**



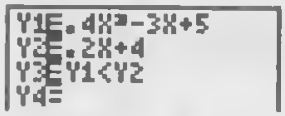
# Graphing Inequalities

**Examine the inequality $.4X^3-3X+5<.2X+4$ graphically. Use the TEST functions to explore the values of X where the inequality is true and where it is false.**

**Procedure**

1. Press [MODE]. Select **DOT**, **SIMUL**, and the default mode settings. Press [2nd] [STAT PLOT], and turn off all stat plots.

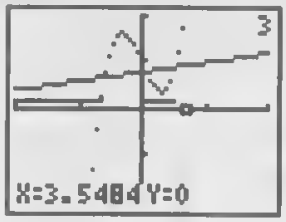
2. Press [Y=], and turn off all functions. Enter the left side of the inequality as **Y1**, the right side as **Y2**, and the statement of the inequality as **Y3**. **Y3** evaluates to 1 if true and 0 if false.



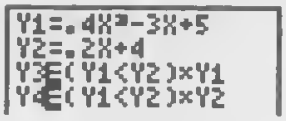

4. Press [WINDOW], and enter these Window variable values.

| | |
|---|---|
| **XMIN=-10** | **YMIN=-10** |
| **XMAX=10** | **YMAX=10** |
| **XSCL=5** | **YSCL=5** |

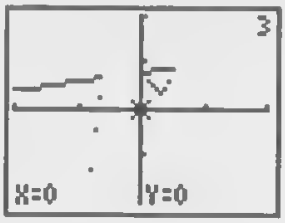
5. Press [TRACE]. Press [▼] [▼] to move to **Y3**. Trace the inequality, observing the value of X. **Y3** is 1 when **Y1< Y2**, and **Y3** is 0 when **Y1 ≥Y2**.




6. Press [Y=] and turn off **Y1** and **Y2**. Enter equations to graph only the inequality.




7. Press [TRACE]. Notice that the values of **Y3** and **Y4** are zero where the inequality is false.



# Graphing a Polar Equation

**The parametric graphing feature of the TI-80 can be used to graph polar equations. Graph the Spiral of Archimedes, the name given to the curve defined by the polar equation r=aθ.**

**Problem**

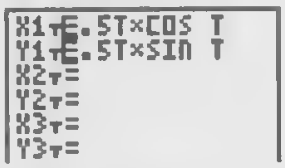
A polar equation r=f(θ) can be graphed by applying the conversion formulae, x=f(θ) cos(θ) and y=f(θ) sin(θ). Thus, the Spiral of Archimedes can be expressed parametrically as:

$x = .5\ \theta \cos(\theta)$
$y = .5\ \theta \sin(\theta)$

**Procedure**

Follow this procedure to solve the problem.

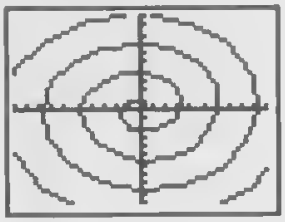
1. Select **PARAM** mode. Choose the defaults for the other mode settings.
2. Enter the expressions to define the parametric equation in terms of T.




3. Set the Window variables to the following values.

| | | |
|---|---|---|
| **TMIN=0** | **XMIN=-10** | **YMIN=-10** |
| **TMAX=25** | **XMAX=10** | **YMAX=10** |
| **TSTEP=π/8** | **XSCL=1** | **YSCL=1** |

4. Press GRAPH to display the Spiral of Archimedes.



# Program: Guess the Coefficients

**This program generates a function in the form A×SIN(BX) with random integer coefficients between 1 and 10. Seven data points from the function are plotted. You are prompted to guess the coefficients, which are plotted as C×SIN(DX). The program continues until your guess is correct. It can be modified for other functions.**

**Program**

```
PROGRAM:GUESS
:FUNC:RADIAN
:-31(π/12)→XMIN
:31(π/12)→XMAX
:π/2→XSCL
:-10→YMIN
:10→YMAX
:1→YSCL
:"A×SIN (BX)"→Y1
:RANDINT(1,10)→A
:RANDINT(1,10)→B
:PLOTSOFF: FNOFF
:DISPGRAPH
:SEQ(XMIN+I×ΔX,I,0,62,9)→L1
:Y1(L1)→L2
:PLOT1(□,L1,L2,□)
:DISPGRAPH
:PAUSE
:"C×SIN (DX)"→Y2
:LBL W
:CLRHOME
:DISP "C×SIN (DX)"
:INPUT "C=",C
:INPUT "D=",D
:DISPGRAPH
:PAUSE
:IF C=A
:DISP "C IS OK"
:IF C>A
:DISP "C IS TOO HIGH"
:IF C<A
:DISP "C IS TOO LOW"
:IF D=B
:DISP "D IS OK"
:IF D>B
:DISP "D IS TOO HIGH"
:IF D<B
:DISP "D IS TOO LOW"
:PAUSE
:IF((C=A)×(D=B))=1
:STOP
:GOTO W
```

Set viewing window.

Initialise coefficients.

Display data points.

Prompt for guesses.

Display results.

Quit if guesses are correct.

# Chapter 12: Memory Management

**This chapter describes how to manage memory on the TI-80. To increase the amount of memory available for use, you may occasionally want to delete stored items that you are no longer using. You can also reset the calculator, erasing all data and programs.**

**Chapter Contents**

# Checking Available Memory

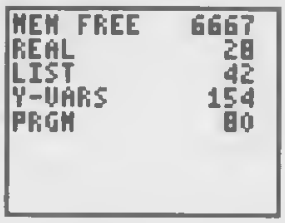
**The MEMORY CHECK RAM screen displays the total amount of memory available and the amount of memory used by each variable type. This allows you to determine the amount of memory available for new items such as programs and the amount used by old items that you no longer need.**

## Displaying the MEM FREE Screen

To check the amounts of available and used memory:

1. Press [2nd] [MEM] to display the **MEMORY** menu.


2. Press 1 or [ENTER] to select **CHECK RAM...**.




   The amount of available memory and the number of bytes used by each variable type are shown on the right.

3. To leave the **CHECK RAM** display:
   - Press [2nd] [QUIT] to go to the Home screen.
   - Press [2nd] [MEM] to return to the **MEMORY** menu.

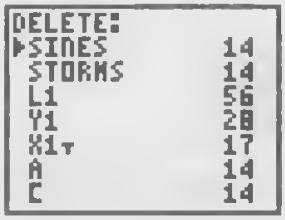
# Deleting Items from Memory

**You can delete the contents of any variable (real number, list, or Y= function) or program from the memory to increase available memory.**

## Deleting an Item

To delete an item:

1. Press [2nd] [MEM] to display the **MEMORY** menu.
2. Press 2 to select **DELETE...** A screen showing all variables currently in use and the amount of memory used by each is displayed.




   The file names are listed in the following order:

   - Program names
   - List names
   - **Y=** equation names
   - Numeric variable names

3. Use [▲] and [▼] to position the cursor (indicated by ▶ in the left column) next to the item you want to delete; and press [ENTER]. The item is deleted immediately.

You can continue to delete individual items from this screen. To leave the **DELETE** display:

- Press [2nd] [QUIT] to go to the Home screen.
- Press [2nd] [MEM] to return to the **MEMORY** menu.

**Note:** Some system variables—**ANS** and statistical variables such as **REGEQ**, for example—cannot be deleted. These system variables are not shown on the **DELETE** display.

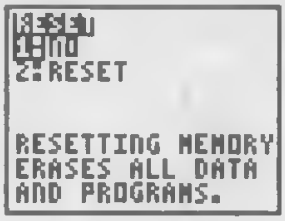
# Resetting the TI-80

**Resetting the TI-80 restores memory to the factory settings, including deleting the contents of all variables and programs and resetting all system variables to the original settings. Because you can increase available memory by deleting individual items, you should rarely need to reset the TI-80.**

## Resetting

To reset the TI-80:

1. Press 2nd [MEM] to display the **MEMORY** menu.
2. Press 3 to select **RESET...**.




3. Make the appropriate menu selection:
   - To go to the Home screen without resetting the memory, select **NO**.
   - To reset the memory, select **RESET**. The Home screen is displayed with the message **MEM CLEARED**.

**Note:** If the screen is blank after **RESET**, adjust the display contrast. Press 2nd and then press and hold ▲ (to make the display darker) or 2nd ▼ (to make the display lighter). You can press CLEAR to clear the message on the display.

# Appendix A: Tables and Reference Information

**This appendix provides a list of all TI-80 functions for use in expressions and instructions that you can use on the Home screen and in programs. It also includes other reference information that can help you.**

**Appendix Contents**

# Table of TI-80 Functions and Instructions

**A function (F) returns a value or a list and can be used in expressions; an instruction (I) initiates an action. Some, but not all, have arguments. † indicates that the instruction is only available for copying from the program editor.**

| | | | |
|---|---|---|---|
| **a⌴b/c** | Sets the display format for fraction results to **a⌴b/c** (mixed fraction) mode. **(I)** | † [MODE] ⟨a⌴b/c⟩ | 1-11 |
| *value*▶**a⌴b/c** | Returns *value* as a mixed fraction. **(F)** | [FRAC] ⟨▶a⌴b/c⟩ | 3-9 |
| **ABS** *value* | Returns the absolute value of *value*. **(F)** | [2nd] [ABS] | 2-6 |
| **ABS** *list* | Returns a list of absolute values for each element in *list*. **(F)** | [2nd] [ABS] | 2-6 |
| Addition: *valueA*+*valueB* | Returns *valueA* plus *valueB*. **(F)** | [+] | 2-4 |
| Addition: *value*+*list* | Returns a list in which *value* is added to each *list* element. **(F)** | [+] | 2-4 |
| Addition: *listA*+*listB* | Returns a list of *listA* elements plus *listB* elements. **(F)** | [+] | 2-4 |
| **AUTOSIMP** | Turns on automatic simplification mode for fractions. **(I)** | † [MODE] ⟨AUTOSIMP⟩ | 1-11 |
| **b/c** | Sets the display format for fraction results to **b/c** (simple fraction) mode. **(I)** | † [MODE] ⟨b/c⟩ | 1-11 |
| *value*▶**b/c** | Returns *value* as a simple fraction. **(F)** | [FRAC] ⟨▶b/c⟩ | 3-9 |
| **CLRDRAW** | Deletes all drawn elements from a graph or drawing. **(I)** | [2nd] [DRAW] DRAW ⟨CLRDRAW⟩ | 7-12 |
| **CLRHOME** | Clears the Home screen. **(I)** | † [PRGM] I/O ⟨CLRHOME⟩ | 10-13 |
| **CLRLIST** *listnameA,listnameB, . . .* | Deletes *listnameA*, *listnameB*, . . .**(I)** | [STAT] EDIT ⟨CLRLIST⟩ | 9-13 |

| | | | |
|---|---|---|---|
| **CONNECTED** | Sets connected line graphing format. **(I)** | † MODE ⟨CONNECTED⟩ | 1-11 |
| **COS** *value* | Returns the cosine of *value*. **(F)** | COS | 2-4 |
| **COS** *list* | Returns a list of the cosine for each *list* element. **(F)** | COS | 2-4 |
| **COS$^{-1}$** *value* | Returns the arccosine of *value*. **(F)** | 2nd [COS$^{-1}$] | 2-4 |
| **COS$^{-1}$** *list* | Returns a list of the arccosine for each *list* element. **(F)** | 2nd [COS$^{-1}$] | 2-4 |
| Cube: *value*$^3$ | Returns the cube of *value*. **(F)** | MATH MATH ⟨$^3$⟩ | 2-8 |
| Cube: *list*$^3$ | Returns a list of the cube for each *list* element. **(F)** | MATH MATH ⟨$^3$⟩ | 2-8 |
| Cube Root: $\sqrt[3]{}$*value* | Returns the cube root of *value*. **(F)** | MATH MATH ⟨$\sqrt[3]{}$⟩ | 2-8 |
| Cube Root: $\sqrt[3]{}$*list* | Returns a list of the cube root for each *list* element. **(F)** | MATH MATH ⟨$\sqrt[3]{}$⟩ | 2-8 |
| *value*▶**DEC** | Returns *value* in decimal form. **(I)** | MATH MATH ⟨▶DEC⟩ | 2-8 |
| *list*▶**DEC** | Returns *list* in decimal form. **(I)** | MATH MATH ⟨▶DEC⟩ | 2-8 |
| **DEGREE** | Sets degree mode. **(I)** | † MODE ⟨DEGREE⟩ | 1-11 |
| Degree Notation: *value*° | Interprets *value* as an angle in degrees. **(F)** | 2nd [ANGLE] ⟨°⟩ | 2-14 |
| **DIM** *list* | Returns the length of *list*. **(F)** | 2nd [LIST] OPS ⟨DIM⟩ | 8-7 |
| *length*→**DIM** *listname* | Creates (if necessary) or redimensions *list* to *length*. **(I)** | 2nd [LIST] OPS ⟨DIM⟩ | 8-7 |
| **DISP** | Displays the Home screen. **(I)** | † PRGM I/O ⟨DISP⟩ | 10-13 |
| **DISP** "*text*" | Displays *text*. **(I)** | † PRGM I/O ⟨DISP⟩ | 10-13 |
| **DISP** *valueA*,*valueB*, . . . | Displays *valueA*, *valueB*, . . .**(I)** | † PRGM I/O ⟨DISP⟩ | 10-13 |
| **DISP** "*text*",*valueA*, "*text*",*valueB*, . . . | Displays *text*,*valueA*,*text* *valueB*, . . .**(I)** | † PRGM I/O ⟨DISP⟩ | 10-13 |

## Table of TI-80 Functions and Instructions (Continued)

| | | | |
|---|---|---|---|
| **DISPGRAPH** | Displays the current graph.**(I)** | † PRGM I/O ⟨DISPGRAPH⟩ | 10-13 |
| Division: *valueA/valueB* | Returns *valueA* divided by *valueB*. **(F)** | ÷ | 2-4 |
| Division: *list/value* | Returns *list* elements divided by *value*. **(F)** | ÷ | 2-4 |
| Division: *value/list* | Returns *value* divided by *list* elements. **(F)** | ÷ | 2-4 |
| Division: *listA/listB* | Returns *listA* elements divided by *listB* elements. **(F)** | ÷ | 2-4 |
| **DOT** | Sets dot graphing format. **(I)** | † MODE ⟨DOT⟩ | 1-11 |
| **DRAWF** *expression* | Draws *expression* (in **X**) on the current graph. **(I)** | 2nd [DRAW] DRAW ⟨DRAWF⟩ | 7-6 |
| **e^***power* | Returns e raised to the value of *power*. **(F)** | 2nd [$e^x$] | 2-5 |
| **e^***list* | Returns a list of e raised to the power of each *list* element. **(F)** | 2nd [$e^x$] | 2-5 |
| **ELSE** *See* **IF:THEN:ELSE** | | | |
| **END** | Identifies the end of a **FOR(**, **IF-THEN**, or **IF-THEN-ELSE** structure. **(I)** | † PRGM CTL ⟨END⟩ | 10-9 |
| Equal: *valueA=valueB* | Returns 1 if *valueA* = *valueB*. Returns 0 if *valueA* ≠ *valueB*. **(F)** | 2nd [TEST] ⟨=⟩ | 2-16 |
| Equal: *listA=listB* | Applies the = test to each element of *listA* and *listB* and returns a list. **(F)** | 2nd [TEST] ⟨=⟩ | 2-16 |
| Equal: *list=value* or *value=list* | Applies the = test to each element of *list* and *value* and returns a list. **(F)** | 2nd [TEST] ⟨=⟩ | 2-16 |
| Exponent: **E***exponent* | Returns 10 to the *exponent*. **(F)** | 2nd [EE] | 1-7 |
| Exponent: *value***E***exponent* | Returns *value* times 10 to the *exponent*. **(F)** | 2nd [EE] | 1-7 |
| Exponent: *list***E***exponent* | Returns *list* elements times 10 to the *exponent*. **(F)** | 2nd [EE] | 1-7 |

| | | | |
|---|---|---|---|
| **EXPREG** *Xlistname*,*Ylistname* | Fits *Xlistname* and *Ylistname* to the exponential model. **(I)** | STAT CALC ⟨EXPREG⟩ | 9–16 |
| **EXPREG** *Xlistname*,*Ylistname*, *freqlistname* | Fits *Xlistname* and *Ylistname* to the exponential model with frequency *freqlistname*.**(I)** | STAT CALC ⟨EXPREG⟩ | 9–16 |
| Factorial: *value***!** | Returns the factorial of *value* ($0 \leq$ integer $\leq 69$). **(F)** | MATH PRB ⟨!⟩ | 2–13 |
| Factorial: *list***!** | Returns a list containing the factorial for each *list* element ($0 \leq$ integers $\leq$ 69). **(F)** | MATH PRB ⟨!⟩ | 2–13 |
| **FIX** *n* | Sets fixed-decimal display mode for *n* decimal places. **(I)** | † MODE ⟨FIX⟩ | 1–10 |
| **FLOAT** | Sets floating-decimal display mode. **(I)** | † MODE ⟨FLOAT⟩ | 1–10 |
| **FNOFF** | Deselects all **Y=** functions. **(I)** | 2nd [Y-VARS] ON/OFF ⟨FNOFF⟩ | 4–8 |
| **FNOFF** *function#*, *function#*, . . .. | Deselects *function#*, *function#*,. . . ..**(I)** | 2nd [Y-VARS] ON/OFF ⟨FNOFF⟩ | 4–8 |
| **FNON** | Selects all **Y=** functions. **(I)** | 2nd [Y-VARS] ON/OFF ⟨FNON⟩ | 4–8 |
| **FNON** *function#*, *function#*, . . . | Selects *function#*, *function#*,. . . .**(I)** | 2nd [Y-VARS] ON/OFF ⟨FNON⟩ | 4–8 |
| **FOR(***variable*,*begin*,*end***)** **:***commands* . . . **:END** | Executes *commands* through **END**, incrementing *variable* from *begin* by 1 until *variable* > *end*. **(I)** | † PRGM CTL ⟨FOR(⟩ | 10–9 |
| **FOR(***variable*,*begin*,*end*, *increment***)** **:***commands* . . . **:END** | Executes *commands* through **END**, incrementing *variable* from *begin* by *increment* until *variable* > *end*. **(I)** | † PRGM CTL ⟨FOR(⟩ | 10–9 |

## Table of TI-80 Functions and Instructions (Continued)

| | | | |
|---|---|---|---|
| **FPART** *value* | Returns the fractional part of *value*. **(F)** | [MATH] NUM ⟨FPART⟩ | 2-10 |
| **FPART** *list* | Returns a list of the fractional parts for each *list* element. **(F)** | [MATH] NUM ⟨FPART⟩ | 2-10 |
| *value*▸**FRAC** | Returns *value* in fraction form, according to the current fraction display format. **(I)** | [FRAC] ⟨▸FRAC⟩ | 3-10 |
| *list*▸**FRAC** | Displays *list* in fraction form, according to the current fraction display format. **(I)** | [FRAC] ⟨▸FRAC⟩ | 3-10 |
| **FUNC** | Sets function graphing mode. **(I)** | † [MODE] ⟨FUNC⟩ | 1-11 |
| **GOTO** *label* | Branches the program to *label*. **(I)** | † [PRGM] CTL ⟨GOTO⟩ | 10-10 |
| Greater Than: *valueA>valueB* | Returns 1 if *valueA* > *valueB*. Returns 0 if *valueA* ≤ *valueB*. **(F)** | [2nd] [TEST] ⟨>⟩ | 2-16 |
| Greater Than: *listA>listB* | Applies the > test to each element of *listA* and *listB* and returns a list. **(F)** | [2nd] [TEST] ⟨>⟩ | 2-16 |
| Greater Than: *list>value* or *value>list* | Applies the > test to each element of *list* and *value* and returns a list. **(F)** | [2nd] [TEST] ⟨>⟩ | 2-16 |
| Greater Than or Equal: *valueA≥valueB* | Returns 1 if *valueA* ≥ *valueB*. Returns 0 if *valueA* < *valueB*.**(F)** | [2nd] [TEST] ⟨≥⟩ | 2-16 |
| Greater Than or Equal: *listA≥listB* | Applies the ≥ test to each element of *listA* and *listB* and returns a list. **(F)** | [2nd] [TEST] ⟨≥⟩ | 2-16 |
| Greater Than or Equal: *list≥value* or *value≥list* | Applies the ≥ test to each element of *list* and *value* and returns a list. **(F)** | [2nd] [TEST] ⟨≥⟩ | 2-16 |
| **GRIDOFF** | Turns the graph grid off. **(I)** | [2nd] [DRAW] DRAW ⟨GRIDOFF⟩ | 4-11 |
| **GRIDON** | Turns the graph grid on. **(I)** | [2nd] [DRAW] DRAW ⟨GRIDON⟩ | 4-11 |

| | | | |
|---|---|---|---|
| **HORIZONTAL** *Y* | Draws a horizontal line at value *Y*. **(I)** | 2nd [DRAW] DRAW ⟨HORIZONTAL⟩ | 7-5 |
| **IF** *condition*:*commandA* :*commands* | If *condition* = 0 (false), skips *commandA*. **(I)** | † PRGM CTL ⟨IF⟩ | 10-7 |
| **IF** *condition* :**THEN**:*commands* :**END** | Executes *commands* from **THEN** to **END** if *condition* = 1 (true). **(I)** | † PRGM CTL ⟨THEN⟩ | 10-8 |
| **IF** *condition* :**THEN**:*commands* :**ELSE**:*commands* :**END** | Executes *commands* from **THEN** to **ELSE** if *condition* = 1 (true); from **ELSE** to **END** if *condition* = 0 (false). **(I)** | † PRGM CTL ⟨ELSE⟩ | 10-8 |
| **INPUT** | Displays the current graph with the free-moving cursor. **(I)** | † PRGM I/O ⟨INPUT⟩ | 10-11 |
| **INPUT** *variable* | Prompts for input to store to *variable*. **(I)** | † PRGM I/O ⟨INPUT⟩ | 10-12 |
| **INPUT** "*text*",*variable* | Prompts using *text* and stores input to *variable*. **(I)** | † PRGM I/O ⟨INPUT⟩ | 10-12 |
| **INT** *value* | Returns the largest integer ≤ *value*. **(F)** | MATH NUM ⟨INT⟩ | 2-10 |
| **INT** *list* | Returns the largest integer ≤ *list* element. **(F)** | MATH NUM ⟨INT⟩ | 2-10 |
| *integerA* **INT÷** *integerB* | Divides *integerA* by *integerB* and returns a quotient (**Q**) and remainder (**R**) on the Home screen, if there are no pending operations. **(F)** | MATH MATH ⟨INT÷⟩ | 2-7 |
| *listA* **INT÷** *listB* | Returns a list of quotients from *listA* and *listB*. **(F)** | MATH MATH ⟨INT÷⟩ | 2-7 |
| *list* **INT÷** *integer* or *integer* **INT÷** *list* | Returns list of quotients from *integer* and *list*. **(F)** | MATH MATH ⟨INT÷⟩ | 2-7 |
| Inverse: $value^{-1}$ | Returns 1 divided by *value*. **(F)** | $x^{-1}$ | 2-4 |
| Inverse: $list^{-1}$ | Returns 1 divided by each *list* element. **(F)** | $x^{-1}$ | 2-4 |

## Table of TI-80 Functions and Instructions (Continued)

| | | | |
|---|---|---|---|
| **IPART** *value* | Returns the integer part of *value*. **(F)** | MATH NUM ⟨IPART⟩ | 2-10 |
| **IPART** *list* | Returns a list of the integer part for each *list* element. **(F)** | MATH NUM ⟨IPART⟩ | 2-10 |
| **LBL** *label* | Assigns *label* to the command. **(I)** | † PRGM CTL ⟨LBL⟩ | 10-10 |
| Less Than: *valueA*<*valueB* | Returns 1 if *valueA* < *valueB*. Returns 0 if *valueA* ≥ *valueB*. **(F)** | 2nd [TEST] ⟨<⟩ | 2-16 |
| Less Than: *listA*<*listB* | Applies the < test to each element of *listA* and *listB* and returns a list. **(F)** | 2nd [TEST] ⟨<⟩ | 2-16 |
| Less Than: *list*<*value* or *value*<*list* | Applies the < test to each element of *list* and *value* and returns a list. **(F)** | 2nd [TEST] ⟨<⟩ | 2-16 |
| Less Than or Equal: *valueA*≤*valueB* | Returns 1 if *valueA* ≤ *valueB*. Returns 0 if *valueA* > *valueB*. **(F)** | 2nd [TEST] ⟨≤⟩ | 2-16 |
| Less Than or Equal: *listA*≤*listB* | Applies the ≤ test to each element of *listA* and *listB* and returns a list. **(F)** | 2nd [TEST] ⟨≤⟩ | 2-16 |
| Less Than or Equal: *list*≤*value* or *value*≤*list* | Applies the ≤ test to each element of *list* and *value* and returns a list. **(F)** | 2nd [TEST] ⟨≤⟩ | 2-16 |
| **LINE(***X1*,*Y1*,*X2*,*Y2*) | Draws line from (*X1*,*Y1*) to (*X2*,*Y2*). **(I)** | 2nd [DRAW] DRAW ⟨LINE(⟩ | 7-4 |
| **LINREG(a+bX)** *Xlistname*,*Ylistname*<br>**LINREG(aX+b)** *Xlistname*,*Ylistname* | Fits *Xlistname* and *Ylistname* to the linear model. **(I)** | STAT CALC ⟨LINREG(a+bX)⟩<br>⟨LINREG(aX+b)⟩ | 9-16<br>9-15 |
| **LINREG(a+bX)** *Xlistname*,*Ylistname*, *freqlistname*<br>**LINREG(aX+b)** *Xlistname*,*Ylistname*, *freqlistname* | Fits *Xlistname* and *Ylistname* to the linear model with frequency *freqlistname*. **(I)** | STAT CALC ⟨LINREG(a+bX)⟩<br>⟨LINREG(aX+b)⟩ | 9-16<br>9-15 |
| **LN** *value* | Returns the natural logarithm of *value*. **(F)** | LN | 2-5 |
| **LN** *list* | Returns a list of the natural logarithm for each *list* element. **(F)** | LN | 2-5 |

| | | | |
|---|---|---|---|
| **LNREG** *Xlistname,Ylistname* | Fits *Xlistname* and *Ylistname* to the logarithmic model. **(I)** | STAT CALC ⟨LNREG⟩ | 9-16 |
| **LNREG** *Xlistname,Ylistname, freqlistname* | Fits *Xlistname* and *Ylistname* to the logarithmic model with frequency *freqlistname*. **(I)** | STAT CALC ⟨LNREG⟩ | 9-16 |
| **LOG** *value* | Returns the logarithm of *value*. **(F)** | LOG | 2-5 |
| **LOG** *list* | Returns a list of the logarithm for each *list* element. **(F)** | LOG | 2-5 |
| **MANSIMP** | Selects manual simplification mode for fractions. **(I)** | † MODE ⟨MANSIMP⟩ | 1-11 |
| **MAX(***valueA,valueB***)** | Returns the larger of *valueA* and *valueB*. **(F)** | 2nd [LIST] MATH ⟨MAX(⟩ | 8-9 |
| **MAX(***list***)** | Returns the largest element in *list*. **(F)** | 2nd [LIST] MATH ⟨MAX(⟩ | 8-9 |
| **MAX(***listA,listB***)** | Returns a list of the larger of each pair of elements in *listA* and *listB*. **(F)** | 2nd [LIST] MATH ⟨MAX(⟩ | 8-9 |
| **MAX(***list,value***)** or **MAX(***value,list***)** | Returns a list of the larger of each *list* element compared to *value*. **(F)** | 2nd [LIST] MATH ⟨MAX(⟩ | 8-9 |
| **MEAN(***list***)** | Returns the mean of *list*. **(F)** | 2nd [LIST] MATH ⟨MEAN(⟩ | 8-9 |
| **MEAN(***list,frequency***)** | Returns the mean of *list* with frequency *frequency*. **(F)** | 2nd [LIST] MATH ⟨MEAN(⟩ | 8-9 |
| **MEDIAN(***list***)** | Returns the median of *list*. **(F)** | 2nd [LIST] MATH ⟨MEDIAN(⟩ | 8-9 |
| **MEDIAN(***list,frequency***)** | Returns the median of *list* with frequency *frequency*. **(F)** | 2nd [LIST] MATH ⟨MEDIAN(⟩ | 8-9 |
| **MIN(***valueA,valueB***)** | Returns the smaller of *valueA* and *valueB*. **(F)** | 2nd [LIST] MATH ⟨MIN(⟩ | 8-9 |
| **MIN(***list***)** | Returns the smallest element in *list*. **(F)** | 2nd [LIST] MATH ⟨MIN(⟩ | 8-9 |

## Table of TI-80 Functions and Instructions (Continued)

| | | | |
|---|---|---|---|
| **MIN(***listA*,*listB*) | Returns a list of the smaller of each pair of elements in *listA* and *listB*. **(F)** | 2nd [LIST] MATH ⟨MIN(⟩ | 8-9 |
| **MIN(***list*,*value*) or **MIN(***value*,*list*) | Returns a list of the smaller of each *list* element compared to *value*. **(F)** | 2nd [LIST] MATH ⟨MIN(⟩ | 8-9 |
| Multiplication: *valueA*×*valueB* | Returns *valueA* times *valueB*. **(F)** | × | 2-4 |
| Multiplication: *value*×*list* or *list*×*value* | Returns a list containing each *value* times each *list* element. **(F)** | × | 2-4 |
| Multiplication: *listA*×*listB* | Returns a list of *listA* elements times *listB* elements. **(F)** | × | 2-4 |
| *valueA* **nCr** *valueB* | Returns the combinations of *valueA* (integer ≥ 0) taken *valueB* (integer ≥ 0) at a time. **(F)** | MATH PRB ⟨nCr⟩ | 2-13 |
| *value* **nCr** *list* | Returns a list of the combinations of *value* (integer ≥ 0) taken each element in *list* (integer ≥ 0) at a time. **(F)** | MATH PRB ⟨nCr⟩ | 2-13 |
| *list* **nCr** *value* | Returns a list of the combinations of each element (integer ≥ 0) in *list* taken *value* (integer ≥ 0) at a time. **(F)** | MATH PRB ⟨nCr⟩ | 2-13 |
| *listA* **nCr** *listB* | Returns a list of the combinations of each element (integer ≥ 0) in *listA* taken each element in *listB* (integer ≥0) at a time. **(F)** | MATH PRB ⟨nCr⟩ | 2-13 |
| **NDERIV(***expression*, *variable*,*value*) | Returns the approximate numerical derivative of *expression* with respect to *variable* at *value*. ε is 1 E −3.**(F)** | MATH MATH ⟨NDERIV(⟩ | 2-9 |
| **NDERIV(***expression*, *variable*,*value*,ε) | Returns the approximate numerical derivative of *expression* with respect to *variable* at *value*, with a specified ε. **(F)** | MATH MATH ⟨NDERIV(⟩ | 2-9 |

| | | | |
|---|---|---|---|
| Negation: *-value* | Returns the negative of *value*. **(F)** | (-) | 2-6 |
| Negation: *-list* | Returns a list with each *list* element negated. **(F)** | (-) | 2-6 |
| **NORMAL** | Sets normal display mode. **(I)** | † MODE ⟨NORMAL⟩ | 1-10 |
| Not Equal: *valueA≠valueB* | Returns 1 if *valueA* ≠ *valueB*. Returns 0 if *valueA* = *valueB*. **(F)** | 2nd [TEST] ⟨≠⟩ | 2-16 |
| Not Equal: *listA≠listB* | Applies the ≠ test to each element of *listA* and *listB* and returns a list. **(F)** | 2nd [TEST] ⟨≠⟩ | 2-16 |
| Not Equal: *list≠value* or *value≠list* | Applies the ≠ test to each element of *list* and *value* and returns a list. **(F)** | 2nd [TEST] ⟨≠⟩ | 2-16 |
| *valueA* **nPr** *valueB* | Returns a list of the permutations of *valueA* (integer ≥ 0) taken *valueB* (integer ≥ 0) at a time. **(F)** | MATH PRB ⟨nPr⟩ | 2-13 |
| *value* **nPr** *list* | Returns a list of the permutations of *value* (integer ≥ 0) taken each element in *list* (integer ≥ 0) at a time. **(F)** | MATH PRB ⟨nPr⟩ | 2-13 |
| *list* **nPr** *value* | Returns a list of the permutations of each element (integer ≥ 0) in *list* taken *value* (integer ≥ 0) at a time. **(F)** | MATH PRB ⟨nPr⟩ | 2-13 |
| *listA* **nPr** *listB* | Returns a list of the permutations of each element (integer ≥ 0) in *listA* taken each element in *listB* (integer ≥ 0) at a time. **(F)** | MATH PRB ⟨nPr⟩ | 2-13 |
| **1-VAR STATS** *listname* | Performs one-variable analysis using *listname* and a frequency of 1. **(I)** | STAT CALC ⟨1-VAR STATS⟩ | 9-15 |
| **1-VAR STATS** *Xlistname*, *freqlistname* | Performs one-variable analysis using *Xlistname* and frequency *freqlistname*. **(I)** | STAT CALC ⟨1-VAR STATS⟩ | 9-15 |
| **PARAM** | Sets parametric graphing mode. | † MODE ⟨PARAM⟩ | 1-11 |

## Table of TI-80 Functions and Instructions (Continued)

| | | | |
|---|---|---|---|
| **PAUSE** | Suspends execution of the program until [ENTER] is pressed. **(I)** | † [PRGM] CTL ⟨PAUSE⟩ | 10-9 |
| **Pi** | Returns the value of π rounded to 13 digits. **(F)** | [2nd] [π] | 2-6 |
| **PLOT***n*(*type*,*Xlist*,*Ylist*) | Plots stat plot *n* (1-3) of *type* (⌐ or ⌐) for *Xlist* and *Ylist* coordinate pairs. | † [2nd] [STAT PLOT] ⟨PLOT*n*⟩ | 9-20 |
| **PLOT***n*(*type*,*Xlist*,*Ylist*, *mark*) | Plots stat plot *n* (1-3) of *type* (⌐ or ⌐) for *Xlist* and *Ylist* coordinate pairs with the specified type of mark. **(I)** | † [2nd] [STAT PLOT] ⟨PLOT*n*⟩ | 9-20 |
| **PLOT***n*(*type*, *Xlist*) or **PLOT***n*(*type*,*Xlist*,*Flist*) | Plots stat plot *n* (1-3) of *type* (⌐ or ⌐) for *Xlist* with frequency *Flist*. If *Flist* is omitted, frequency = 1. **(I)** | † [2nd] [STAT PLOT] ⟨PLOT*n*⟩ | 9-20 |
| **PLOTSOFF** | Deselects all stat plots. **(I)** | [2nd] [STAT PLOT] ⟨PLOTSOFF⟩ | 9-21 |
| **PLOTSOFF** *plot#,plot#. . .* | Deselects stat *plot1*, *plot2*, or *plot3*. **(I)** | [2nd] [STAT PLOT] ⟨PLOTSOFF⟩ | 9-21 |
| **PLOTSON** | Selects all stat plots. **(I)** | [2nd] [STAT PLOT] ⟨PLOTSON⟩ | 9-21 |
| **PLOTSON** *plot#,plot#. . .* | Selects stat *plot1*, *plot2*, or *plot3*. **(I)** | [2nd] [STAT PLOT] ⟨PLOTSON⟩ | 9-21 |
| Power: *value*^*power* | Returns *value* raised to *power*. **(F)** | [^] | 2-5 |
| Power: *list*^*power* | Returns a *list* of each element raised to the value of *power*. **(F)** | [^] | 2-5 |
| Power: *value*^*list* | Returns a list of *value* raised to the power of each *list* element. **(F)** | [^] | 2-5 |
| Power: *listA*^*listB* | Returns a list of each *listA* element raised to the power of each *listB* element. **(F)** | [^] | 2-5 |
| Power of ten: **10**^*power* | Returns 10 raised to the value of *power*. **(F)** | [2nd] [$10^x$] | 2-5 |
| Power of ten: **10**^*list* | Returns a list of 10 raised to the power of each *list* element. **(F)** | [2nd] [$10^x$] | 2-5 |

| | | | |
|---|---|---|---|
| **PRGM**_*programname* | Executes the program *programname*. **(I)** | † PRGM CTRL ⟨PRGM_(⟩ | 10-10 |
| **PROD** *list* | Returns the product of elements in *list*. **(F)** | 2nd [LIST] MATH ⟨PROD⟩ | 8-10 |
| **P▸Rx(***R*,θ**)** | Returns the rectangular coordinate **x**, given the polar coordinates *R* and θ. **(F)** | 2nd [ANGLE] ⟨P▸Rx(⟩ | 2-15 |
| **P▸Rx(***Rlist*,θ**)** | Returns a list of **x** coordinates, given the *R* coordinates in *Rlist* and a single θ. **(F)** | 2nd [ANGLE] ⟨P▸Rx(⟩ | 2-15 |
| **P▸Rx(***R*,θ*list***)** | Returns a list of **x** coordinates, given the single *R* coordinate and the θ coordinates in θ*list*. **(F)** | 2nd [ANGLE] ⟨P▸Rx(⟩ | 2-15 |
| **P▸Rx(***Rlist*,θ*list***)** | Returns a list of **x** coordinates, given the *R* and θ coordinates in *Rlist* and θ *list*. **(F)** | 2nd [ANGLE] ⟨P▸Rx(⟩ | 2-15 |
| **P▸Ry(***R*,θ**)** | Returns rectangular coordinate **y**, given polar coordinates *R* and θ. **(F)** | 2nd [ANGLE] ⟨P▸Ry(⟩ | 2-15 |
| **P▸Ry(***Rlist*,θ**)** | Returns a list of **y** coordinates, given the *R* coordinates in *Rlist* and a single θ coordinate. **(F)** | 2nd [ANGLE] ⟨P▸Ry(⟩ | 2-15 |
| **P▸Ry(***R*,θ*list***)** | Returns a list of **y** coordinates, given a single *R* coordinate and the θ coordinates in θ*list*. **(F)** | 2nd [ANGLE] ⟨P▸Ry(⟩ | 2-15 |
| **P▸Ry(***Rlist*,θ*list***)** | Returns a list of **y** coordinates, given the *R* coordinates in *Rlist* and the θ coordinates in θ*list*. **(F)** | 2nd [ANGLE] ⟨P▸Ry(⟩ | 2-15 |
| **PT-CHANGE(***X*,*Y***)** | Toggles the point at (*X*,*Y*). **(I)** | 2nd [DRAW] POINTS ⟨PT-CHANGE(⟩ | 7-10 |
| **PT-OFF(***X*,*Y***)** | Erases the point at (*X*,*Y*). **(I)** | 2nd [DRAW] POINTS ⟨PT-OFF(⟩ | 7-10 |
| **PT-ON(***X*,*Y***)** | Draws the point at (*X*,*Y*). **(I)** | 2nd [DRAW] POINTS ⟨PT-ON(⟩ | 7-10 |

## Table of TI-80 Functions and Instructions (Continued)

| | | | |
|---|---|---|---|
| **PWRREG** *Xlistname, Ylistname* | Fits *Xlistname* and *Ylistname* to the power model. **(I)** | [STAT] CALC ⟨PWRREG⟩ | 9-16 |
| **PWRREG** *Xlistname, Ylistname,freqlistname* | Fits *Xlistname* and *Ylistname* to the power model with frequency *freqlistname*. **(I)** | [STAT] CALC ⟨PWRREG⟩ | 9-16 |
| **QUADREG** *Xlistname, Ylistname* | Fits *Xlistname* and *Ylistname* to the quadratic model. **(I)** | [STAT] CALC ⟨QUADREG⟩ | 9-16 |
| **QUADREG** *Xlistname, Ylistname,freqlistname* | Fits *Xlistname* and *Ylistname* to the quadratic model with frequency *freqlistname*. **(I)** | [STAT] CALC ⟨QUADREG⟩ | 9-16 |
| **RADIAN** | Sets radian mode. **(I)** | † [MODE] ⟨RADIAN⟩ | 1-11 |
| Radian Notation: *value*ʳ | Interprets *value* as an angle in radians. | [2nd] [ANGLE] ⟨ʳ⟩ | 2-14 |
| **RAND** | Returns a random number between 0 and 1. **(F)** | [MATH] PRB ⟨RAND⟩ | 2-12 |
| **RANDINT(***lower, upper***)** | Returns a random integer between *lower* and *upper*. **(F)** | [MATH] PRB ⟨RANDINT(⟩ | 2-13 |
| **RANDINT(***lowerlist,upper***)** | Returns a list of random integers between each element of *lowerlist* and *upper*. **(F)** | [MATH] PRB ⟨RANDINT(⟩ | 2-13 |
| **RANDINT(***lower,upperlist***)** | Returns a list of random integers between *lower* and each element of *upperlist*. **(F)** | [MATH] PRB ⟨RANDINT(⟩ | 2-13 |
| **RANDINT(***lowerlist, upperlist***)** | Returns a list of random integers between each element of *lowerlist* and *upperlist*. **(F)** | [MATH] PRB ⟨RANDINT(⟩ | 2-13 |
| **REMAINDER(***valueA, valueB***)** | Returns the remainder from the division of *valueA* by *valueB* | [MATH] NUM ⟨REMAINDER(⟩ | 2-11 |
| **REMAINDER(***value, list***)** | Returns a list of remainders from the division of *value* by each element in *list*. | [MATH] NUM ⟨REMAINDER(⟩ | 2-11 |

| | | | |
|---|---|---|---|
| **REMAINDER(***list*, *value*) | Returns a list of remainders from the division of each element in *list* by *value*. | MATH NUM ⟨REMAINDER(⟩ | 2-11 |
| **REMAINDER(***listA*, *listB*) | Returns a list of remainders from the division of each element in *listA* by the each element in *listB*. | MATH NUM ⟨REMAINDER(⟩ | 2-11 |
| **RETURN** | Returns to the calling program. **(I)** | † PRGM CTL ⟨RETURN⟩ | 10-10 |
| $n^{th}root\sqrt[x]{value}$ | Returns $n^{th}root$ of *value*. **(F)** | MATH MATH ⟨$\sqrt[x]{}$⟩ | 2-8 |
| $n^{th}root\sqrt[x]{list}$ | Returns a list of $n^{th}root$ for each *list* element. **(F)** | MATH MATH ⟨$\sqrt[x]{}$⟩ | 2-8 |
| $list\sqrt[x]{value}$ | Returns *list* roots of *value*. **(F)** | MATH MATH ⟨$\sqrt[x]{}$⟩ | 2-8 |
| $listA\sqrt[x]{listB}$ | Returns a list of *listA* roots of *listB*. **(F)** | MATH MATH ⟨$\sqrt[x]{}$⟩ | 2-8 |
| **ROUND(***value*) | Returns *value* rounded to 10 digits. **(F)** | MATH NUM ⟨ROUND(⟩ | 2-10 |
| **ROUND(***value*,*#decimals*) | Returns *value* rounded to *#decimals* (≤ 9). **(F)** | MATH NUM ⟨ROUND(⟩ | 2-10 |
| **ROUND(***list*) | Returns *list* elements rounded to 10 digits. **(F)** | MATH NUM ⟨ROUND(⟩ | 2-10 |
| **ROUND(***list*,*#decimals*) | Returns *list* elements rounded to *#decimals* (≤ 9). **(F)** | MATH NUM ⟨ROUND(⟩ | 2-10 |
| **R▸Pr(***X*,*Y*) | Returns the polar coordinate **r**, given the rectangular coordinates *X* and *Y*. **(F)** | 2nd [ANGLE] ⟨R▸Pr(⟩ | 2-15 |
| **R▸Pr(***Xlist*,*Y*) | Returns a list of **r** coordinates, given the *X* coordinates in *Xlist* and a single *Y* coordinate. **(F)** | 2nd [ANGLE] ⟨R▸Pr(⟩ | 2-15 |
| **R▸Pr(***X*,*Ylist*) | Returns a list of **r** coordinates, given a single *X* coordinate and the *Y* coordinates in *Ylist*. **(F)** | 2nd [ANGLE] ⟨R▸Pr(⟩ | 2-15 |

| | | | |
|---|---|---|---|
| **R▸Pr(***Xlist*,*Ylist*) | Returns a list of **r** coordinates, given the *X* coordinates in *Xlist* and the *Y* coordinates in *Ylist*. **(F)** | 2nd [ANGLE] ⟨R▸Pr(⟩ | 2-15 |
| **R▸Pθ(***X*,*Y*) | Returns the polar coordinate θ, given the rectangular coordinates *X* and *Y*. **(F)** | 2nd [ANGLE] ⟨R▸Pθ(⟩ | 2-15 |
| **R▸Pθ(***Xlist*,*Y*) | Returns a list of θ coordinates, given the *X* coordinates in *Xlist* and a single *Y* coordinate. **(F)** | 2nd [ANGLE] ⟨R▸Pθ(⟩ | 2-15 |
| **R▸Pθ(***X*,*Ylist*) | Returns a list of θ coordinates, given a single *X* coordinate and the *Y* coordinates in *Ylist*. **(F)** | 2nd [ANGLE] ⟨R▸Pθ(⟩ | 2-15 |
| **R▸Pθ(***Xlist*,*Ylist*) | Returns a list of θ coordinates, given the *X* coordinates in *Xlist* and the *Y* coordinates in *Ylist*. **(F)** | 2nd [ANGLE] ⟨R▸Pθ(⟩ | 2-15 |
| **SCI** | Sets scientific display mode. **(I)** | † MODE ⟨SCI⟩ | 1-10 |
| **SEQ(***expression*,*variable*, *begin*,*end*,*increment*) | Returns a list created by evaluating *expression* for *variable*, from *begin* to *end* by *increment*. **(F)** | 2nd [LIST] OPS ⟨SEQ(⟩ | 8-8 |
| **SEQUENTIAL** | Sets sequential graphing mode. **(F)** | † MODE ⟨SEQUENTIAL⟩ | 1-11 |
| **SHADE(***lowerfunc*, *upperfunc*) | Shades the area above *lowerfunc* and below *upperfunc*. **(I)** | 2nd [DRAW] DRAW ⟨SHADE(⟩ | 7-9 |
| **SHADE(***lowerfunc*, *upperfunc*,*resolution*) | Shades the area above *lowerfunc*, below *upperfunc*, and with *resolution* (1 to 9). **(I)** | 2nd [DRAW] DRAW ⟨SHADE(⟩ | 7-9 |
| **SHADE(***lowerfunc*, *upperfunc*,*resolution*, *Xleft*) | Shades the area above *lowerfunc*, below *upperfunc*, to right of **X**=*Xleft*, and with *resolution* (1 to 9). **(I)** | 2nd [DRAW] DRAW ⟨SHADE(⟩ | 7-9 |

| | | | |
|---|---|---|---|
| **SHADE(***lowerfunc*, *upperfunc*,*resolution*, *Xleft*,*Xright*) | Shades the area above *lowerfunc*, below *upperfunc*, to right of **X**=*Xleft*, to left of **X**=*Xright*, and with *resolution* (1 to 9). **(I)** | 2nd [DRAW] DRAW ⟨SHADE(⟩ | 7-9 |
| **SHADE_Y>***func1*,*func2*,. . . | Shades the area above *func1* with vertical pattern and above *func2* with diagonal pattern (lower left to upper right), etc. **(I)** | 2nd [DRAW] DRAW ⟨SHADE_Y>⟩ | 7-7 |
| **SHADE_Y<***func3*,*func4*,. . . | Shades the area below *func3* with horizontal pattern and below *func4* with diagonal pattern (upper left to lower right), etc. **(I)** | 2nd [DRAW] DRAW ⟨SHADE_Y<⟩ | 7-8 |
| *fraction*▶**SIMP** | Simplifies *fraction* by its lowest common factor. **(F)** | FRAC ⟨SIMP⟩ | 3-8 |
| (*fraction*,*factor*)▶**SIMP** | Simplifies *fraction* by the specified factor, which must be an integer. **(F)** | FRAC ⟨SIMP⟩ | 3-8 |
| **SIMUL** | Sets simultaneous graphing mode. **(I)** | † MODE ⟨SIMUL⟩ | 1-11 |
| **SIN** *value* | Returns the sine of *value*. **(F)** | SIN | 2-4 |
| **SIN** *list* | Returns a list of the sine for each *list* element. **(F)** | SIN | 2-4 |
| **SIN$^{-1}$** *value* | Returns the arcsine of *value*. **(F)** | 2nd [SIN$^{-1}$] | 2-4 |
| **SIN$^{-1}$** *list* | Returns a list of the arcsine for each *list* element. **(F)** | 2nd [SIN$^{-1}$] | 2-4 |
| **SORTA(***listname*) | Sorts *listname* elements in ascending order. **(I)** | 2nd [LIST] OPS ⟨SORTA(⟩ | 8-6 |
| **SORTA(***keylistname*, *dependlist1*, *dependlist2*,...) | Sorts the elements of *keylistname* in ascending order with *dependlist1*, *dependlist2* . . . as dependent lists. **(I)** | 2nd [LIST] OPS ⟨SORTA(⟩ | 8-6 |

## Table of TI-80 Functions and Instructions (Continued)

| | | | |
|---|---|---|---|
| **SORTD(***listname***)** | Sorts the elements of *listname* in descending order. **(I)** | 2nd [LIST] OPS ⟨SORTD()⟩ | 8-6 |
| **SORTD(***keylistname, dependlist1, dependlist2,...***)** | Sorts the elements of *keylistname* in descending order with *dependlist1,dependlist2* . . . as dependent lists. **(I)** | 2nd [LIST] OPS ⟨SORTD()⟩ | 8-6 |
| Square: $value^2$ | Returns *value* multiplied by itself. **(F)** | $x^2$ | 2-5 |
| Square: $list^2$ | Returns a list of each *list* element squared. **(F)** | $x^2$ | 2-5 |
| Square Root: $\sqrt{}value$ | Returns the square root of *value*. **(F)** | 2nd [$\sqrt{}$] | 2-5 |
| Square Root: $\sqrt{}list$ | Returns a list of the square roots of each *list* element. **(F)** | 2nd [$\sqrt{}$] | 2-5 |
| **STOP** | Stops program execution and returns to the Home screen. **(I)** | † PRGM CTL ⟨STOP⟩ | 10-10 |
| Store: *value*→*variable* | Stores *value* to *variable*. **(I)** | STO▸ | 1-13 |
| Store: *list*→*listname* | Stores *list* to *listname*. **(I)** | STO▸ | 1-13 |
| Store: "*expression*"→*Yn* or "*expression*"→*XnT* or "*expression*"→*YnT* | Stores *expression* to *Yn* *XnT* or *YnT* function. **(I)** | STO▸ | 1-13 |
| Subtraction: *valueA*−*valueB* | Subtracts *valueB* from *valueA*. **(F)** | − | 2-4 |
| Subtraction: *value*−*list* | Subtracts *list* elements from *value* and returns a list. **(F)** | − | 2-4 |
| Subtraction: *list*−*value* | Subtracts *value* from *list* elements and returns a list. **(F)** | − | 2-4 |
| Subtraction: *listA*−*listB* | Subtracts each *listB* element from each *listA* element and returns a list. **(F)** | − | 2-4 |
| **SUM** *list* | Returns the sum of elements in *list*. **(F)** | 2nd [LIST] MATH ⟨SUM⟩ | 8-10 |
| **TAN** *value* | Returns the tangent of *value*. **(F)** | TAN | 2-4 |

| | | | |
|---|---|---|---|
| **TAN** *list* | Returns a list of the tangent for each *list* element. **(F)** | TAN | 2-4 |
| **TAN$^{-1}$** *value* | Returns the arctangent of *value*. **(F)** | 2nd [TAN$^{-1}$] | 2-4 |
| **TAN$^{-1}$** *list* | Returns a list of the arctangent for each *list* element. **(F)** | 2nd [TAN$^{-1}$] | 2-4 |
| **THEN**<br>*See* **IF:THEN** | | | |
| **TRACE** | Displays a graph and enters Trace mode. **(I)** | † TRACE | 4-13 |
| **2-VAR STATS**<br>*Xlistname*,*Ylistname* | Performs two-variable analysis using *Xlistname* and *Ylistname*. **(I)** | STAT CALC ⟨2-VAR STATS⟩ | 9-15 |
| **2-VAR STATS**<br>*Xlistname*,*Ylistname*,<br>*freqlistname* | Performs two-variable analysis using *Xlistname* and *Ylistname* with frequency *freqlistname*. **(I)** | STAT CALC ⟨2-VAR STATS⟩ | 9-15 |
| **VERTICAL** *X* | Draws a vertical line at value *X*. **(I)** | 2nd [DRAW] DRAW ⟨VERTICAL⟩ | 7-5 |
| **ZBOX** | Displays a graph to allow the user to define new viewing window. **(I)** | † ZOOM ⟨ZBOX⟩ | 4-15 |
| **ZDECIMAL** | Displays a graph in new viewing window. **(I)** | † ZOOM ⟨ZDECIMAL⟩ | 4-17 |
| **ZOOM IN** | Displays a graph in new viewing window. **(I)** | † ZOOM ⟨ZOOM IN⟩ | 4-16 |
| **ZOOM OUT** | Displays a graph in new viewing window. **(I)** | † ZOOM ⟨ZOOM OUT⟩ | 4-16 |
| **ZSQUARE** | Displays a graph in new viewing window. **(I)** | † ZOOM ⟨ZSQUARE⟩ | 4-17 |
| **ZSTANDARD** | Displays a graph in new viewing window. **(I)** | † ZOOM ⟨ZSTANDARD⟩ | 4-17 |
| **ZTRIG** | Displays a graph in new viewing window. **(I)** | † ZOOM ⟨ZTRIG⟩ | 4-17 |

# TI-80 Menu Map

**Menus begin in the upper left of the keyboard. Default values are shown.**

[Y=]

| (**FUNC** mode) | (**PARAM** mode) |
|---|---|
| Y1= | X1T= |
| Y2= | Y1T= |
| Y3= | X2T= |
| Y4= | Y2T= |
| | X3T= |
| | Y3T= |

[WINDOW]

| (**FUNC** mode) | (**PARAM** mode) |
|---|---|
| WINDOW | WINDOW |
| XMIN=-10 | TMIN=0 |
| XMAX=10 | TMAX=2π |
| XSCL=1 | TSTEP=π/24 |
| YMIN=-10 | XMIN=-10 |
| YMAX=10 | XMAX=10 |
| YSCL=1 | XSCL=1 |
| | YMIN=-10 |
| | YMAX=10 |
| | YSCL=1 |

[2nd] [TblSet]

```
TABLE SETUP
 TBLMIN=0
 ΔTBL=1
```

[ZOOM]

```
ZOOM
1:ZBOX
2:ZOOM IN
3:ZOOM OUT
4:ZDECIMAL
5:ZSQUARE
6:ZSTANDARD
7:ZTRIG
```

[MODE]

```
NORMAL SCI
FLOAT 0123456789
RADIAN DEGREE
a␣b/c b/c
AUTOSIMP MANSIMP
FUNC PARAM
CONNECTED DOT
SEQUENTIAL SIMUL
```

[MODE] (in program editor)

```
MODE
1:NORMAL
2:SCI
3:FLOAT
4:FIX
5:RADIAN
6:DEGREE
7:a␣b/c
8:b/c
9:AUTOSIMP
0:MANSIMP
A:FUNC
B:PARAM
C:CONNECTED
D:DOT
E:SEQUENTIAL
F:SIMUL
```

2nd [STAT PLOT]

```
STAT PLOTS
1:PLOT1…
  OFF ⌐ L1 L2 ▫
2:PLOT2…
  OFF ⌐ L1 L2 ▫
3:PLOT3…
  OFF ⌐ L1 L2 ▫
4:PLOTSOFF
5:PLOTSON
```

2nd [STAT PLOT] (in program editor)

```
PLOTS       TYPE    MARK
1:PLOT1(    1:⌐     1:▫
2:PLOT2(    2:⌐     2:+
3:PLOT3(    3:≡     3:•
4:PLOTSOFF  4:▥
5:PLOTSON
```

Screen for ≡ or ▥ plots
(1-variable plots)

```
PLOTn
ON OFF
TYPE: ⌐ ⌐ ≡ ▥
XL: L1L2L3L4L5L6
F: 1L1L2L3L4L5L6
```

Screen for ⌐ or ⌐ plots
(2-variable plots)

```
PLOTn
ON OFF
TYPE: ⌐ ⌐ ≡ ▥
XL: L1L2L3L4L5L6
YL: L1L2L3L4L5L6
MARK: ▫ + •
```

STAT

```
EDIT          CALC
1:EDIT…       1:1-VAR STATS
2:SORTA(      2:2-VAR STATS
3:SORTD(      3:LINREG(aX+b)
4:CLRLIST     4:QUADREG
              5:LINREG(a+bX)
              6:LNREG
              7:EXPREG
              8:PWRREG
```

[2nd] [LIST]

| OPS | MATH |
|---|---|
| 1:SORTA( | 1:MIN( |
| 2:SORTD( | 2:MAX( |
| 3:DIM | 3:MEAN( |
| 4:SEQ( | 4:MEDIAN( |
| | 5:SUM |
| | 6:PROD |

[MATH]

| MATH | NUM | PRB |
|---|---|---|
| 1:INT+ | 1:ROUND( | 1:RAND |
| 2:▶DEC | 2:IPART | 2:nPr |
| 3:$^3$ | 3:FPART | 3:nCr |
| 4:$\sqrt[3]{}$ | 4:INT | 4:! |
| 5:$\sqrt[x]{}$ | 5:MIN( | 5:RANDINT( |
| 6:NDERIV( | 6:MAX( | |
| | 7:REMAINDER( | |

[FRAC]

| FRACTION |
|---|
| 1:▶SIMP |
| 2:▶b/c |
| 3:▶a⌴b/c |
| 4:▶FRAC |
| 5:▶DEC |

[2nd] [TEST]

| TEST |
|---|
| 1:= |
| 2:≠ |
| 3:> |
| 4:≥ |
| 5:< |
| 6:≤ |

[2nd] [ANGLE]

| ANGLE |
|---|
| 1:° |
| 2:r |
| 3:R▶Pr( |
| 4:R▶Pθ( |
| 5:P▶Rx( |
| 6:P▶Ry( |

PRGM

| EXEC | EDIT | NEW |
|---|---|---|
| 1:*name* | 1:*name* | 1:CREATE NEW |
| 2:*name* | 2:*name* | |
| 3:*name* | 3:*name* | |
| ⋮ | ⋮ | |

PRGM (in program editor)

| CTL | I/O | EXEC |
|---|---|---|
| 1:IF | 1:INPUT | 1:*name* |
| 2:THEN | 2:DISP | 2:*name* |
| 3:ELSE | 3:DISPGRAPH | 3:*name* |
| 4:FOR( | 4:CLRHOME | ⋮ |
| 5:END | | |
| 6:PAUSE | | |
| 7:LBL | | |
| 8:GOTO | | |
| 9:PRGM_ | | |
| 0:RETURN | | |
| A:STOP | | |

2nd [DRAW]

| DRAW | POINTS |
|---|---|
| 1:CLRDRAW | 1:PT-ON( |
| 2:LINE( | 2:PT-OFF( |
| 3:HORIZONTAL | 3:PT-CHANGE( |
| 4:VERTICAL | |
| 5:DRAWF | |
| 6:SHADE_Y> | |
| 7:SHADE_Y< | |
| 8:SHADE( | |
| 9:GRIDON | |
| 0:GRIDOFF | |

[VARS]

VARS
1:WINDOW…
2:STATISTICS…
3:TABLE…
4:SIMPFACTOR…

[VARS] ⟨WINDOW…⟩

| X/Y | T |
|---|---|
| 1:XMIN | 1:TMIN |
| 2:XMAX | 2:TMAX |
| 3:XSCL | 3:TSTEP |
| 4:YMIN | |
| 5:YMAX | |
| 6:YSCL | |
| 7:ΔX | |
| 8:ΔY | |
| 9:XFACT | |
| 0:YFACT | |

[VARS] ⟨TABLE…⟩

TABLE
1:TBLMIN
2:ΔTBL

[VARS] ⟨SIMPFACTOR…⟩

⑤⑤⑤⑤

SIMPFACTOR
1:FACTOR

[VARS] ⟨STATISTICS…⟩

| X/Y | Σ | EQ | BOX |
|---|---|---|---|
| 1:n | 1:ΣX | 1:a | 1:Q1 |
| 2:$\bar{x}$ | 2:ΣX$^2$ | 2:b | 2:MED |
| 3:Sx | 3:ΣY | 3:c | 3:Q3 |
| 4:σx | 4:ΣY$^2$ | 4:r | |
| 5:$\bar{y}$ | 5:ΣXY | 5:REGEQ | |
| 6:Sy | | | |
| 7:σy | | | |
| 8:MINX | | | |
| 9:MAXX | | | |
| 0:MINY | | | |
| A:MAXY | | | |

[2nd] [Y-VARS]

| Y | XT/YT | ON/OFF |
|---|---|---|
| 1:Y1 | 1:X1T | 1:FNON |
| 2:Y2 | 2:Y1T | 2:FNOFF |
| 3:Y3 | 3:X2T | |
| 4:Y4 | 4:Y2T | |
| | 3:X3T | |
| | 4:Y3T | |

2nd [MEM]

```
MEMORY
1:CHECK RAM…
2:DELETE…
3:RESET…
```

2nd [MEM] ⟨CHECK RAM…⟩

```
MEM FREE  7014
REAL        14
LIST         0
Y-VARS      80
PRGM        14
```

2nd [MEM] ⟨DELETE…⟩

```
DELETE:
▶name    memory
 name    memory
 name    memory
  ⋮        ⋮
```

(*names* include defined programs, lists, Y= equations, and user variables, in that order.)

2nd [MEM] ⟨RESET…⟩

```
1:NO
2:RESET

RESETTING MEMORY
ERASES ALL DATA
AND PROGRAMS.
```

**The variables listed below are used by the TI-80 in various ways. Some have restrictions on their use.**

### User Variables

The variables **A** to **Z** and θ can hold only numbers—either decimals or fractions. You may store to these variables. The TI-80 can update **X**, **Y**, and **T** during graphing; therefore, you may wish to reserve those variables for graphing activities.

The variables **L1** to **L6** are defined as lists. You cannot store another type of data to them.

You can store any string of characters, functions, instructions, or variable names to the functions **Y*n*** (in **FUNC** mode) and **X*n*T** and **Y*n*T** (in **PARAM** mode), by using either [STO▸] or the **Y=** editor. The validity of the string is determined when the function is evaluated.

### System Variables

The values of the Window variables—**XMIN**, **XMAX**, **XSCL**, **ΔX**, **TSTEP**, etc.—must be real numbers. You may store to them. Since the TI-80 can update some of them, as the result of a Zoom instruction, for example, you may wish to reserve these variables for graphing activities.

The statistical result variables—**n**, $\bar{x}$, **MINX**, **ΣX**, **a**, **r**, **REGEQ**, **X1**, **Y1**, **Q1**, **MED**, **Q3**, etc.—are reserved for use by the TI-80. You cannot store to them.

# Appendix B: Service and Warranty Information

**This appendix provides supplementary information that may be helpful as you use the TI-80. It includes procedures that may help you correct problems with the calculator, and it describes the service and warranty provided by Texas Instruments.**

**Appendix Contents**

# Battery Information

**The TI-80 uses two CR2032 lithium 3-volt batteries.**

### When to Replace the Batteries

As you use the TI-80, the battery voltage will gradually drop, and the display will dim. You can adjust the contrast to darken the display when this happens. If the display is dim and adjusting the contrast to level 9 does not make it dark enough, you should replace the batteries. Refer to pages B-3 to B-7 for instructions on how to change the batteries.

### Battery Precautions

Follow these safety guidelines concerning batteries.

- Do not mix new and used batteries.
- Do not mix different types of battery.
- Follow polarity diagrams carefully.
- Do not replace batteries with rechargeable batteries.
- Do not place non-rechargeable batteries in a battery charger.
- Properly dispose of used batteries immediately. Do not leave them within the reach of small children.
- Do not incinerate used batteries.

### Retaining Stored Data

**Caution:** The TI-80 retains stored data when you are changing batteries only if you:

- Do not remove both batteries at the same time. (At least one battery must be installed at all times in order to retain memory.)
- Turn the unit off and do not turn it back on until you have changed the batteries.
- Do not allow the batteries to run down completely before changing them.

Follow the steps on pages B-3 to B-7 when changing batteries.

**Changing the Batteries (Continued)**

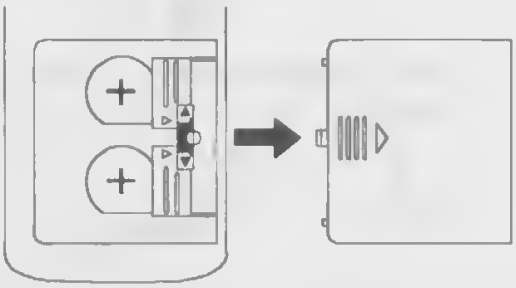
To change the batteries, first:

a. **Turn the calculator off.**

b. Replace the plastic slide cover over the keys.

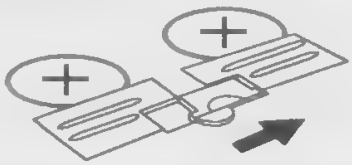
c. Turn the calculator so that the back is facing you.

1.



Placing your thumb on the ridged area of the plastic battery compartment cover, push down slightly and slide the cover about 1/4-inch to the right. You can then lift the cover off.

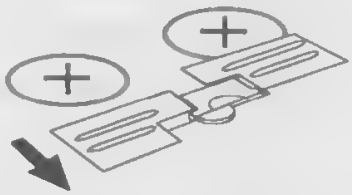
2.



Push the red switch up to free the metal battery cover over the lower battery.

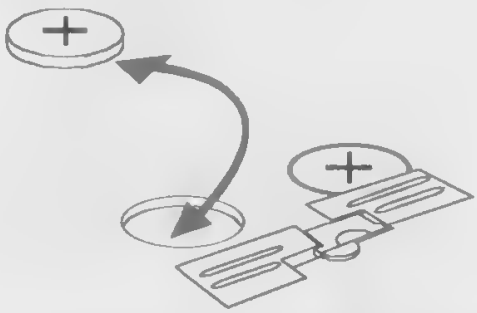
**Changing the Batteries (Continued)**

3.



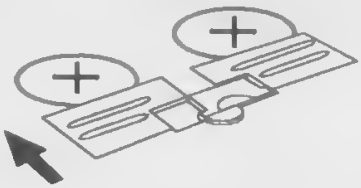
Slide the metal battery cover away from the battery. The battery pops up.

4.



Remove the old battery. Insert a new battery, positive side (+) up.

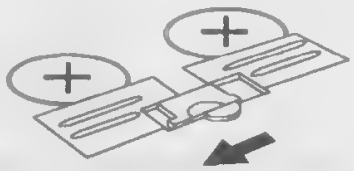
5.



Hold the new battery in place and slide the metal battery cover back into position over the battery.

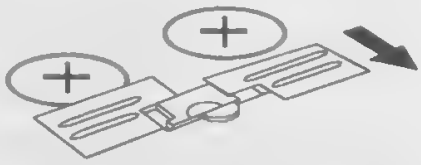
**Changing the Batteries (Continued)**

6.



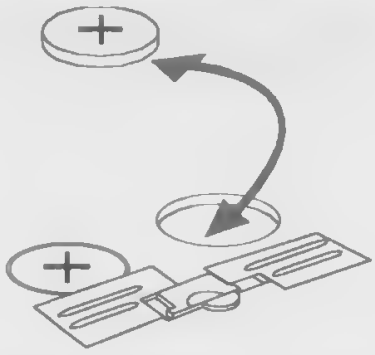
Push the red switch all the way down to free the metal battery cover over the upper battery.

7.



Slide the metal battery cover to the right.

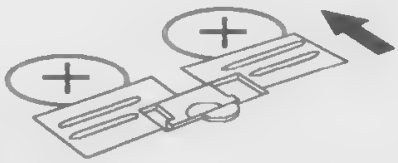
8.



Remove the old battery. Insert a new battery, positive side (+) up.

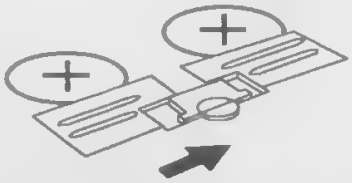
**Changing the Batteries (Continued)**

9.



Hold the new battery in place and slide the metal battery cover back into position over the battery.

10.

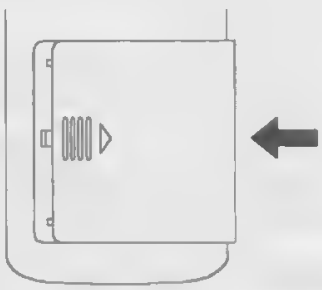


Push the red switch to its centre position to lock the metal covers and batteries into place.

**Note:** The calculator will not turn on unless the red switch is in the centre position.

**Changing the Batteries (Continued)**

11.



Replace the plastic battery compartment cover. Turn the calculator on and adjust the display contrast.

To adjust the display contrast, press and release the [2nd] key. To increase the contrast (darken the screen), press and hold [▲]. To decrease the contrast (lighten the screen), press and hold [▼].

**To maximise accuracy, the TI-80 carries more digits internally than it displays.**

### Computational Accuracy

Values in memory are stored using up to 13 digits with a two-digit exponent.

- You can store values in the Window variables using up to 10 digits (13 digits for **XSCL**, **YSCL**, and **TSTEP**).
- When a value is displayed, the displayed value is rounded as specified by the **MODE** setting (Chapter 1), with a maximum of 10 digits and a two-digit exponent.
- **REGEQ** displays up to 13 digits.

### Graphing Accuracy

**XMIN** is the centre of the left-hand pixel, **XMAX** is the centre of the next to the right-hand pixel. (The right-hand pixel is reserved for the busy indicator.) **ΔX** is the distance between the centres of two adjacent pixels.

- **ΔX** is calculated as (**XMAX–XMIN**)/**62**.
- If **ΔX** is entered from the Home screen or a program, then **XMAX** is calculated as **XMIN**+**ΔX**×**62**.

**YMIN** is the centre of the next to the bottom pixel, **YMAX** is the centre of the top pixel. **ΔY** is the distance between the centres of two adjacent pixels.

- **ΔY** is calculated as (**YMAX-YMIN**)/**46**.
- If **ΔY** is entered from the Home screen or a program, then **YMAX** is calculated as **YMIN**+**ΔY**×**46**.

Cursor coordinates are displayed as six characters, which may include a negative sign, decimal point, and exponent.

## Function Limits

Below is a table of functions and the range of input values for each.

| Function | Input Values |
|---|---|
| SIN $x$, COS $x$, TAN $x$ | $0 \leq \lvert x \rvert < 10^{10}$ (degree) |
| $SIN^{-1}$ $x$, $COS^{-1}$ $x$ | $-1 \leq x \leq 1$ |
| LN $x$, LOG $x$ | $10^{-100} < x < 10^{100}$ |
| $e^x$ | $-10^{100} < x \leq 230.2585092993$ |
| $10^x$ | $-10^{100} < x < 100$ |
| $\sqrt{x}$ | $0 \leq x < 10^{100}$ |
| $x!$ | $0 \leq x \leq 69$, where $x$ is an integer |

## Function Results

Below is a table of functions and the range of the result for each.

| Function | Range of Result |
|---|---|
| $SIN^{-1}$ $x$, $TAN^{-1}$ $x$ | $-90°$ to $90°$ or $-\pi/2$ to $\pi/2$ (radians) |
| $COS^{-1}$ $x$ | $0°$ to $180°$ or 0 to $\pi$ (radians) |

**If you have difficulty operating the calculator, the following suggestions may help you to correct the problem.**

## Handling a Difficulty

Follow these procedures if you have difficulties.

1. If you cannot see anything on the display, perhaps the display contrast needs adjusting.

   Press and release the [2nd] key. To increase the contrast (darken the screen), press and hold [▲]. To decrease the contrast (lighten the screen), press and hold [▼].

   You will find additional information on display contrast on page I-3.

2. If after adjusting the display contrast, the calculator does not appear to be working at all, ensure the batteries are installed properly and that they are fresh. Refer to "Battery Information" beginning on page B-2 for more details.

   **Note:** Make sure the red switch in the battery compartment is in the centre position.

3. If an error occurs, follow the procedure on page 1-22. Refer to the more detailed explanations about specific errors beginning on page B-11, if necessary.

4. If the cursor is a checked pattern, memory is full. Press [2nd] [MEM] **DELETE...** and delete some items from memory. See Chapter 12 for additional information about memory management.

5. If the dotted-line busy indicator is displayed, a graph or program is paused, and the TI-80 is waiting for input. Press [ENTER] to continue or [ON] to break.

**When the TI-80 detects an error, it displays ERR:*message* and the error menu. The general procedure for correcting errors is described on page 1-22. The error messages, their possible causes, and suggestions for correction are shown below.**

| | |
|---|---|
| **ARGUMENT** | A function or instruction does not have the correct number of arguments. See Appendix A and the appropriate chapter. |
| **BREAK** | You have pressed the [ON] key to break execution of a program, halt a Draw instruction, or stop evaluation of an expression. |
| **DATA TYPE** | You have entered a value or variable of the wrong data type.<br>• A function (including implied multiplication) or an instruction has an argument of an invalid data type; for example, a list where a real number is required. See Appendix A and the appropriate chapter.<br>• You are attempting to store to an incorrect data type; for example, a list to a real variable.<br>• In function graphing or parametric graphing, you have generated a list result rather than a single value; for example, attempting to graph **Y1={1,2,3}×X**. |
| **DIM MISMATCH** | You are attempting to perform an operation that uses more than one list, but the dimensions do not match. |
| **DOMAIN** | Typically, this occurs when the value of an argument does not fall within a specified range.<br>• You are attempting to divide by zero.<br>• You are attempting a logarithmic or power regression with a **-X** or an exponential or power regression with a **-Y**.<br>• A zero value for ε for **NDERIV(** will result in this error.<br>This error does not occur during graphing because the TI-80 allows for undefined values on a graph. |

| | |
|---|---|
| **INCREMENT** | • The increment in **SEQ(** is **0** or has the wrong sign. This error does not occur during graphing. The TI-80 allows for undefined values on a graph.<br>• The increment in **FOR(** is **0** or has the wrong sign. |
| **INVALID** | You are attempting to reference a variable or use a function in a place where it is not valid. For example, **Y*n*** cannot reference **Y**, **XMIN**, **ΔX**, or **TBLMIN**. |
| **INVALID DIM** | • The dimension of the argument is not appropriate for the operation.<br>• List element dimensions must be integers between 1 and 99; for example, **L1(100)** will cause an error. |
| **LABEL** | The label in the **GOTO** instruction is not defined with a **LBL** instruction in the program. |
| **MEMORY** | • There is insufficient memory in which to perform the desired command. You must delete item(s) from memory (Chapter 12) before executing this command.<br>• Using an **IF/THEN** or **FOR(** with a **GOTO** that branches out of the loop can also cause this error because the **END** statement that terminates the loop is never reached. |
| **MODE** | You are attempting a **▶SIMP** in **AUTOSIMP** mode. |
| **NEST LEVEL** | This error occurs when any nested combination of function evaluation, **NDERIV(** or **SEQ(** exceeds 5 levels. |
| **OVERFLOW** | You are attempting to enter, or you have calculated, a number that is beyond the range of the calculator. This error does not occur during graphing. The TI-80 allows for undefined values on a graph. |

| | |
|---|---|
| **STAT** | • You are attempting a linear regression with a vertical line.<br>• Statistical analyses must have at least two data points; **QUADREG** must have at least three data points.<br>• The list of **F** (frequency) elements must be ≥ 0, and at least one **F** value must be ≥ 0.<br>• The frequency list, when used for "sorting" statistics (median, $Q_1$, $Q_3$, or boxplot), must be an integer ≥ 0 and ≤ 99.<br>• **(XMAX-XMIN)/XSCL** must be ≤ 31 for a histogram. |
| **STAT PLOT** | You are trying to display a graph when there is a statistical plot turned on that uses an undefined list. |
| **SYNTAX** | The command contains a syntax error. Look for misplaced functions, arguments, parentheses, or commas. See Appendix A and the appropriate chapter. |
| **UNDEFINED** | You are attempting to reference a variable that is not currently defined. For example, a statistical variable , which has no current value because a list has been edited, has been referenced. |
| **WINDOW RANGE** | There is a problem with the Window variables.<br>• You may have defined **XMAX≤XMIN**, **YMAX≤YMIN**, **TSTEP=0**, or **TMAX≤TMIN** and **TSTEP>0** (or vice versa).<br>• The Window variables are too small or too large to graph correctly. This can occur if you attempt to zoom in or out so far that you are not within the numerical range of the calculator. |
| **ZOOM** | A point or a line, rather than a box, is defined in **ZBOX**; or a math error has resulted from a Zoom operation. |

# Two-Year Contractual Warranty

1. The terms and conditions set out hereinunder shall not apply where you have purchased this calculator directly from Texas Instruments Ltd. in which the conditions of sale of Texas Instruments Ltd. shall apply.

2. This electronic calculator (including charger if applicable) from Texas Instruments is warranted to the original purchaser for a period of two (2) years from the original purchase date - normal use and service - against defective materials or workmanship. For those calculators designed to incorporate batteries, this warranty does not cover damage resulting from any battery leakage. Batteries delivered with calculators are for demonstration purposes only. This warranty is void if: the calculator has been damaged by accident or other causes not arising out of defects in material or workmanship.

   During the above two-year period, the calculator or its defective parts will be repaired, adjusted and/or replaced with a reconditioned model of equivalent quality, ("RECONDITIONED") at manufacturer's option without charge to the purchaser when the calculator is returned, by way of the dealer to Texas Instruments with proof-of-purchase date. UNITS RETURNED WITHOUT PROOF OF PURCHASE DATE WILL BE RETURNED AT THE CARRIAGE RATES IN EFFECT AT THE TIME OF RETURN.

   In the event of replacement with a reconditioned model, the replacement unit will continue to be covered by the warranty of the original calculator product or for a period of 90 days, whichever is longer.

   THIS CONDITION 2 SHALL NOT AFFECT THE STATUTORY RIGHTS OF A CONSUMER AS DEFINED IN THE CONSUMER TRANSACTIONS (RESTRICTIONS ON STATEMENTS) ORDER 1976 (AS AMENDED).

3. Save as expressly provided in Condition 2, Texas Instruments shall be under no liability of whatsoever kind, howsoever caused whether or not due to the negligence or willful default of Texas Instruments or its servants or agents arising out of or in connection with this calculator provided that nothing contained in this condition 3 shall exclude or restrict:

   (I) Any liability of Texas Instruments for death or personal injury resulting from the negligence of Texas Instruments or its servants or agents; or

   (II) Any liability of Texas Instruments for loss or damage arising from this calculator (within the meaning of Sec. 5 9(2) (A) Unfair Contract Terms Act. 1977) and resulting from the negligence of Texas Instruments or its servants or agents.

TEXAS INSTRUMENTS

TI-80 User Manual

GB

TEXAS INSTRUMENTS

Printed in Holland
Verweij B.V. Mijdrecht
Communicatieweg 1
3641 SG Mijdrecht



TI80/OM-GB
PN1113355-0001